

Reference

ADC-1

Owner's Manual

Remote Measurement Systems, Inc.

2633 Eastlake Avenue East

P.O. Box 15544 • ~~6233-39th Avenue N.E.~~

Seattle, Washington ~~98115~~ 98102

(206) ~~525-3369~~

328-2255

June 1983

©Copyright Remote Measurement Systems, Inc., 1983

All Rights Reserved. Specifications and information
subject to change without prior notification.

Rev. 84-1

TABLE OF CONTENTS

Warranty	3
Warnings	4
1.0 Introduction	5
2.0 Getting Started	7
2.1 Connecting the ADC-1 to Your Computer	7
Cabling, Ground, Power, Data Lines,	
Baud Rates, Stop Bits, Parity	
2.2 Operating the ADC-1	10
Testing the A/D by Reading	
an Analog Input	
Reading a Digital Input	
Testing the Control of	
AC Line Carrier Modules	
Testing the Controlled Outputs	
3.0 Programming for the ADC-1	15
3.1 Command Byte Structure	15
3.2 Response Byte Structure	16
3.3 Testing Communications	18
3.4 Programming the Analog to Digital Converter	
to Read the Onboard Temperature Sensor	20
3.5 Selecting and Reading Any Analog Input Channel	21
3.6 Programming for Digital Input Operation	21
3.7 Program Control of AC Line Carrier Modules	23
3.8 Programming the Controlled Outputs	25
4.0 Details	27
4.1 Communications	27
4.2 Analog to Digital Converter	28
Analog Inputs	
Instrumentation Amplifier	
4.3 Digital Inputs	31
4.4 BSR Control	33
4.5 Controlled Outputs	33
4.6 Applications Using the ADC-1 Board Select	34
4.7 Powering the ADC-1 from Batteries	34
5.0 Analog and Digital Sensor Descriptions	29
5.1 Installation and Operation	35
5.2 Temperature Sensors	36
Solid State Sensors	
Thermocouples	

5.3	Wind Speed Sensors	39
5.4	Electrical Energy Consumption Sensor	40
5.5	Light Level Sensors	41
5.6	Optical Encoders	41
5.7	Security System Sensors	42
	Magnetic Switches	
	Vibration Sensor	
	Floor Mat Pressure Sensor	
	Infrared and Ultrasonic Detectors	
5.8	DC Voltage Scaling	43
6.0	Applications	45

Appendixes

A.	Alternate Protocols for ADC-1/Computer	
	Interconnection	A-1
	VIC 20/Commodore 64	
	Atari 400/800	
	S-100 Z80A using SSM IO-4	
	NorthStar Horizon	
	Osborne 1 and Executive	
	Epson HX-20	
B.	Basic Programs For Specific Computers	B-1
	VIC-20/Commodore 64	
	Atari 400/800	
	IBM Personal Computer	
	Osborne 1 and Executive	
	Epson HX-20	
	A Data Acquisition and Control System	
	with Terminal Display and Graphing	
C.	Communication Codes	C-1
	C-1 Command and Response Bytes -	
	Bit Assignments	
	C-2 ADC-1 Command Instruction Set	
	C-3 BSR House Code Switch Settings	
	C-4 BSR AC Module Control Codes	
D.	Datasheets	D-1
	D-1 Intersil 7109 Analog to Digital Converter	
	D-2 Intersil 6402 UART	
	D-3 Intersil 7605 Instrumentation Amplifier	
	D-4 Analog Devices AD590 Temperature Transducer	
	D-4 Amprobe A603 AC Current Transducer	
E.	Specifications	E-1
	E-1 ADC-1 Technical Specifications	
	E-2 Pin Assignments for the DB-25 Connector	
	E-3 Parts List	
	E-4 Component Layout Diagram	
F.	References	F-1

Warranty

Remote Measurement Systems, Inc. (RMSI) warrants that its products will be free from defects in material and workmanship for a period of 6 months from the date of purchase. In the event of malfunction or other indication of failure attributable directly to faulty workmanship and/or material, then, upon return of the ADC-1 to Remote Measurement Systems, Inc., 6233-39th Ave. N.E., Seattle, Washington, 98115, RMSI will, at its option, repair or replace the defective part or parts to restore said product to proper operating condition. All such repairs and/or replacements shall be rendered by RMSI without charge for parts or labor when the product is returned within the specified period from the date of purchase. If an ADC-1 is received that has been damaged in shipment, contact the carrier and process a claim.

This warranty will not cover the failure of Remote Measurement System products, which at the discretion of RMSI shall have resulted from accident, abuse, negligence, unauthorized alteration, or misapplication of the product. While every effort had been made to provide clear and accurate technical information on the application of the ADC-1, RMSI assumes no liability in any events which may arise from the use of said technical information.

THIS WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING WARRANTIES OF MERCHANTABILITY AND FITNESS FOR USE. IN NO EVENT WILL REMOTE MEASUREMENT SYSTEMS, INC. BE LIABLE FOR THE INCIDENTAL AND CONSEQUENTIAL DAMAGES ARISING FROM OR IN ANY WAY CONNECTED WITH THE USE OF ITS PRODUCTS. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION MAY NOT APPLY TO YOU. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS AND YOU MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM STATE TO STATE.

Warnings

IN NO EVENT SHOULD 110 VAC CIRCUITS BE CONNECTED DIRECTLY TO THE ADC-1. ANY AND ALL INTERCONNECTIONS WITH 110 VAC CIRCUITS SHOULD BE DONE IN ACCORDANCE WITH LOCAL REGULATIONS BY COMPETENT ELECTRICAL WORKERS. USE EXTREME CAUTION IN WORKING WITH 110 VOLT CIRCUITS AS SERIOUS INJURY OR DEATH MAY RESULT.

THE CONTROL OF EXTERNAL DEVICES SHOULD BE WELL PLANNED, AND IMPLEMENTED WITH CAUTION. THE EFFECTS OF A FAILURE OF POWER SUPPLY, COMPONENTS, OR WIRING, OR OTHER ACCIDENTAL OCCURANCES SHOULD BE CONSIDERED AND DETERMINED NOT TO ADVERSELY IMPACT THE INSTALLATION. DEVICES UNDER PROGRAM CONTROL SHOULD NOT BE LOCATED WHERE THEIR ACTIVATION COULD INJURE BYSTANDERS OR OTHERS UNAWARE OF IMPENDING ACTIVITY.

For example, do not allow your computer to control a fan in a location where a child could possibly be playing with the apparently "off" fan. Any use of computer controlled heaters should be evaluated for fire danger. The ADC-1 has not been designed nor certified for use in critical life support functions.

THERE ARE NO USER SERVICABLE PARTS WITHIN THE SMALL BLUE PLASTIC BSR TRANSMITTER ENCLOSURE. HOWEVER, SHOULD YOU DESIRE TO OPEN THIS BOX, REMOVE THE PLUG FROM THE AC OUTLET BEFORE OPENING IN ORDER TO AVOID ELECTRIC SHOCK.

1.0 Introduction

The Remote Measurement Systems' ADC-1 is a low cost, reliable and easy to operate data acquisition and control system. The ADC-1 provides a computer with the capability of sensing and reacting to information gathered from the surrounding environment. The ADC-1 functions by measuring a sensor's analog voltage output and converting this to a digital number which can be evaluated by the computer. Under conditions which can be programmed in any computer language, the ADC-1 can be employed to control external devices. You do not need to have sophisticated electronic skills, since the ADC-1 supplies the computer/sensor/control interface. With an ADC-1, you are free to concentrate on the design and installation of the sensors required by your specific application.

The main features of the ADC-1 are:

- * 16 analog input channels coupled with a 12 bit analog to digital converter (A/D) permitting the ADC-1 to perform precision measurements.
- * 4 digital inputs can be used to monitor or count on/off signals.
- * 6 controlled TTL outputs allow the operation of external electronic devices.
- * Remote control of appliances and lamps via signals transmitted over AC wiring to BSR modules.
- * Optional instrumentation amplifier providing microvolt resolution.
- * RS-232 communications for easy interconnection to most computers or modems.

This manual is organized to provide the reader with increasingly detailed descriptions of ADC-1 operation in each successive chapter. It is suggested that the manual be initially scanned in its entirety, noting the sections that correspond to your understanding and requirements. A person with minimal experience in programming or hardware installation should find the initial examples easy to follow and implement. Those users with professional skills and knowledge will find more complex control and measurement capabilities of the ADC-1 described in later chapters.

Chapter 2 provides information for connecting the ADC-1 to a computer. A specific instruction sequence is provided to test each functional component of the ADC-1. The steps demonstrate the functionality of the ADC-1 without going into detail.

Chapter 3 describes the operation and capabilities of the ADC-1. Each feature is demonstrated by short programs that illustrate proper communication protocols with the ADC-1. The examples implement basic control and measurement programs that can be used as building blocks for the creation of sophisticated applications programs.

Chapter 4 details the hardware capabilities that are available for more complex control and measurement applications.

Chapter 5 contains a description of sensors that have been employed with the ADC-1 and contains instructions on their installation.

Chapter 6 provides a look at the various applications for the ADC-1.

The Appendices contain installation notes for specific systems, programing examples, ADC-1 command summaries, technical datasheets, and references.

With the installation of your ADC-1, your computer will have the ability not only to measure and sense its environment, but to control external devices. However, the control is only as good as the programming and sensor/control installation. The user must carefully plan any system implementing the control features to avoid accidents. Remote Measurement Systems cannot be liable for damages resulting from the use of the ADC-1.

Remote Measurement Systems would like to remain responsive to the needs of its customers. All comments and suggestions regarding this manual and the ADC-1 will be carefully considered. Please take the time to complete the enclosed registration form so that you will receive product and manual updates.

In addition, Remote Measurement Systems is serving as a clearing house for user developed software that may be applicable to other ADC-1 owners. A database of applications and software availability is maintained as an aid to new users.

2.0 Getting Started

By now you have probably picked up and examined your ADC-1. The larger blue aluminum box contains the printed circuit board that performs most of the functions of the ADC-1. On the top surface are two sets of terminal strips to which sensor or control cables may be attached with a screwdriver. Each block of terminals is labeled as to its function and each terminal position is numbered.

The smaller box, which plugs into an alternating current(AC) outlet, is the transmitter for modulating signals on the AC wiring which control the remotely located modules. The transmitter was placed in a separate enclosure to isolate AC voltages from the sensitive A/D circuitry. A small flat 3 wire cable runs from the transmitter box to a DB-25 male (with pins) connector. This connector mates with the female DB-25 on the right-hand side panel of the ADC-1 enclosure and will serve as the ADC-1 end of the cable to your computer.

2.1 Connecting the ADC-1 to Your Computer

Communication with the ADC-1 requires four wires be connected between the RS-232 connector on your computer and the DB-25 on the ADC-1. One wire connects the ground, the second provides power, while the remaining two wires are used for transmitting and receiving data.

In order to prepare the cable to connect the ADC-1 to your computer, you will need the following items:

- * a connector which mates with your computer's serial port connector and forms one end of the computer to ADC-1 cable.
- * a short length of four conductor wire.
- * a soldering iron and solder.
- * screwdriver, pliers, and wire stripper/cutter.
- * owner's manual for your computer.

You may need to refer to your computer's instruction manual to determine which pins to connect on your computer. The pin layout of the DB-25 male connector provided with the ADC-1 is described in Appendix E-2.

If you would prefer to have Remote Measurement Systems prepare a cable for you, please furnish us with the required length, computer type, and complete pin designations of your I/O port.

Cabling

The type of wire used to connect the ADC-1 with your computer is not critical. Shielded twisted pair cabling may be needed for operation at separations of greater than 20 feet. Shorter distances may not require shielding and inexpensive twisted pair can be used. Flexibility and durability of your cable will be greater if the wire is stranded rather than solid.

The DB-25 male connector attached to the AC signal transmitter serves as the ADC-1 end of the cable to the your computer. Pins 11, 12, and 13 are already occupied by the flat cable to the BSR transmitter. Connection of your computer cabling to the DB-25 is accomplished by soldering the required four wires into the open pins.

Ground

Pin 7 on the DB-25 connector of the ADC-1 serves as both the signal and power ground and should be connected to your computer signal ground. Pin 7 is the standard ground pin for a RS-232C serial port.

Power

An internal power supply was not designed into the ADC-1 in order to reduce the cost. CMOS (complementary metal oxide semiconductor) chips were used throughout to minimize power consumption. The ADC-1 electronics consume about 60 mW at 5 volts. External sensors may consume an additional 100 mw with terminal strips provided with the ADC-1.

The power source should be capable of supplying 20ma at +5 volts, or 35ma at from +6 to +24 volts. This power can generally be obtained from the computer or modem connected to the ADC-1. Most computer power supplies are designed to supply more than 1 amp so the addition of 20 ma. should not cause overloading for most AC powered computers.

In many cases power can be obtained from the RS-232 outputs of your computer. Some serial IO boards have pins for either +5 or +12 volts clearly stated for limited currents. Alternately, connection may be made directly with the power supply of your computer or with a separate external supply. Be sure to use a DC supply as the ADC-1 uses a zener diode to regulate the board voltage and may function unpredictably if a noisy power supply is used.

In remote locations or for portable applications, the ADC-1 may be powered by batteries. A complete discussion of battery interconnection is contained in section 4.7.

On the ADC-1 DB-25 connector, pin 18 should be used for connection of a regulated +5 volts, or current limited +12 volts. Pin 9 should be used for supply voltages exceeding 6 volts but less than 25 volts. If you have a voltmeter, you can check that the ADC-1 is receiving power correctly by testing the voltage between the terminals labeled "+5 V @ 5ma" and "GND" on the left of the ADC-1. You should measure a voltage between 4.7 and 5.8 volts.

WARNING! Switching the ground and power lines, or application of more than 5.6 volts with sufficient current (150 ma.) between pin 7 and 21 can damage the ADC-1. Further discussion of the consequences and repair procedures can be found in section 4.1.

Data Lines

On the ADC-1, data is received from your computer on pin 3 and transmitted to your computer on pin 2 of the DB-25 connector. Thus TRANSMITTED DATA from your computer is connected to RECEIVED DATA on the ADC-1 and RECEIVED DATA on your computer is connected to TRANSMITTED DATA on the ADC-1. If in the following examples the ADC-1 fails to communicate with your computer it is likely that you have incorrectly connected RECEIVED and TRANSMITTED DATA. Switch the two lines and try again.

Instructions for interconnecting the ADC-1 with specific computers and modems may be found in Appendix A.

Baud Rates

The ADC-1 is capable of communication at baud rates of 300, 600, 1200, 2400, 4800, and 9600. Operation at 9600 baud reduces the time needed to communicate with the ADC-1 thus increasing measurement and control speed.

Baud rates are set on the ADC-1 by changing switch positions on the under side of the ADC-1 enclosure. The baud rate switches (SW2) are in the box center and are numbered from 1 to 6, corresponding to the six selectable baud rates. Both the ADC-1 and controlling computer or modem must be set to identical baud rates. To set the ADC-1 baud rate, turn on the switch corresponding to the desired

baud rate listed below. Closing more than one switch on SW-2 positions 1 through 6 will result in improper operation. The ADC-1 comes shipped to you with the baud rate switches set for 300 baud.

SW-2	Switch	No.	Selected	Baud Rate
	1	-----		2400
	2	-----		4800
	3	-----		600
	4	-----		1200
selected	→ 5	-----		9600
	6	-----		300

Stop Bits

The ADC-1 is hard wired to send and receive two stop bits. However, it functions for most RS-232 computer ports successfully with one stop bit if you wait for the ADC-1 response prior to sending another command. If possible set your RS-232 port for 2 stop bits.

Parity

The ADC-1 uses all eight bits of each byte for information and requires that no parity be selected on your computer port. Check your computer or modem manual for directions.

2.2 Operating the ADC-1

Once connection of the ADC-1 to your computer or modem has been achieved then you can proceed to test the unit. In this section the functional components of the ADC-1 are tested using a sequence of one line commands entered from your terminal. Though not all of the ADC-1 capabilities are utilized, by the end of this section you should be able to say "Hey, it works!"

All commands to the ADC-1 take the form of 8-bit bytes. For each byte sent to the ADC-1, it will respond by sending a byte back to the host computer. For all commands sent to the ADC-1 the returned character indicates that a command was received and processed. The information content of the response depends on the command sent. Refer to section 3.1 for a discussion of ADC-1 commands.

For the purposes of explanation and instruction, the operation of the ADC-1 is discussed with examples using BASIC. Any language which can send data to and from an external device can be used, including FORTRAN, PASCAL, and your system's assembly language. The test examples of the following sections utilize the Microsoft BASIC commands INP and OUT to access the serial port connected to the ADC-1. These commands are specific to 8080 and Z-80 microprocessor based systems where input/output port (I/O) addressing is independent of memory. However, computers using the 6502 microprocessor such as the Apple, VIC-20/Commodore 64, or Atari, use memory addresses to access I/O ports. To properly execute the following tests you must determine which serial I/O port number or memory location in your computer is connected to the ADC-1. More information on I/O ports may be found in section 3.3.

For systems using memory addressed I/O ports make the following substitutions for each example:

- 1) Substitute "PEEK(m1)" for "INP(pn)"
- 2) Substitute "POKE m1,j" for "OUT pn,j"

where: m1 - memory location of the I/O port
 pn - I/O port number
 j - integer interpreted as a command
 by the ADC-1 (0 - 255 range).

The following pages contain step by step illustrations of the operation of the ADC-1. Note that these steps cannot be entered as a executable program because they do not check the status of the input port to determine if the response byte from the ADC-1 has been returned. Typing each instruction is sufficiently slow so that the ADC-1 response byte is assured to have been received by the computer before the next command is typed into the terminal.

Testing the Analog to Digital Converter

Since the ADC-1 is shipped with an internal AD590 temperature sensor, it is possible to check the A/D operation by reading the onboard temperature. Type in each command without a line number. End each command with a RETURN from the computer terminal.

To read the voltage of a sensor attached to one of the analog inputs on the ADC-1, change the value 24 in step 1 to the analog input number on the terminal strip minus one. The analog inputs on the terminal strip are numbered 1 to 16, while the commands selecting the corresponding analog channel range from 0 to 15. The remainder of the steps remain the same.

<u>Step</u>	<u>Command</u>	<u>Description</u>
a)	OUT pn,24	This tells the ADC-1 to connect the output of the temperature sensor to the A/D converter and start the conversion process.
b)	OUT pn,160	The ADC-1 is commanded to send back the high byte from the A/D converter. Since the ADC-1 has a 12-bit A/D, it requires two 8-bit bytes to return the full digital value.
c)	Y=INP(pn)	This reads the A/D high byte into your computer from the serial port and stores it in Y.
d)	OUT pn,144	The ADC-1 is commanded to send back the low byte from the A/D.
e)	X=INP(pn)	The low byte is read from the computer serial port and assigned to variable X.
f)	Z=X+256*(Y AND 15)	Four bits from the high byte are masked, scaled and added to the low byte.
g)	IF (Y AND 16) THEN Z=Z ELSE Z=-Z	The sign bit is examined and the reading is given the proper sign.
h)	PRINT Z/10	The value printed is the output of the temperature sensor in millivolts. This number is the temperature in degrees Kelvin of the uncalibrated AD 590 sensor. It should be within 6 degrees of the true temperature. Convert to Fahrenheit by the formula $F=9/5*(K-273.2)+32$.

Reading a Digital Input

The following sequence of steps will test the operation of digital input #1.

<u>Step</u>	<u>Command</u>	<u>Description</u>
a)	OUT pn,128	This command resets the latched digital inputs (#1 and #3) to zero.
b)	OUT pn,129	Tells the ADC-1 to send the status of the digital inputs.
c)	X=INP(pn)	Read the digital inputs status from the computer serial port.
d)	PRINT (X AND 1)	Check to see if input is now zero.
e)	To change the logic state of input #1, briefly touch a short piece of wire between the positive and negative sides of the digital input terminal labeled "1". Repeat steps 2, 3, and 4. The value printed in step d should now be one.	

For a full discussion of the digital inputs including an explanation of latching and the use of digital inputs in counting pulses, refer to sections 3.6 and 4.3.

Testing the Control of AC Line Carrier Modules

To test the ADC-1 AC line carrier control you need a BSR control module. These modules are manufactured by BSR (USA) Ltd. and are sold by a number of distributors including Sears and Radio Shack under the brand names BSR, Leviton, and Plug n' Power. They are all equivalent. We will subsequently refer to any of these units as BSR modules.

On the BSR module, you will see two rotary dials. Set the house code letter to "A", and then set the unit number to "1". Plug the module into an AC outlet and plug a lamp into the module. Be sure the lamp is turned on so that you will be able to see the results of your computer generated commands. Also be sure the transmitter box is plugged into an AC outlet.

- a) OUT pn,204 This command tells the ADC-1 to send a code on the AC lines which informs any module set with an "A1" to get ready to receive an order.
- b) OUT pn,197 The lamp plugged into the module should now be turned on.
- c) OUT pn,199 The lamp should now go out.

Section 3.7 explains the BSR system and contains a program which will cause your lamp to "count".

Testing the Controlled Outputs

The controlled outputs are TTL logic levels and can be directed to be either ON(+5v) or OFF(0v). When ON, they supply up to 50 ma @ 5 volts. Please refer to section 4.5 for limitations on the power handling capabilities of the ADC-1 outputs. In this small test, you can check the operation of the outputs with a voltmeter. When the output is on, it should read +5 volts; when off it should read 0 volts.

- 1) OUT pn,64 This commands the ADC-1 to place all 6 controlled outputs in the OFF(0v) condition. You may test this with your meter by checking the voltage across each pair of outputs on the lower right-hand side of the ADC-1.
- 2) OUT pn,127 The ADC-1 controlled outputs should now be all ON (5v). Again test this with your voltmeter.

A program to control individual outputs is shown in section 3.8.

3.0 Programming for the ADC-1

The capabilities of the ADC-1 are best demonstrated by placing the ADC-1 under the control of your computer. In this chapter, the details of programming for the ADC-1 are presented by using sample programs to illustrate the operation of ADC-1 features.

3.1 Command Byte Structure

Each command and response from the ADC-1 occupies an 8-bit byte in which the value of each bit has an operational meaning. The bit placement within a byte is defined as follows:

	MSB							LSB
Bit Number	8	7	6	5	4	3	2	1
Decimal Value	128	64	32	16	8	4	2	1

Once the value of each of the eight bits has been selected, the byte may be converted to a decimal number which can be sent to the RS232 port via the BASIC "OUT(I,J)" or "POKE(m)" commands.

The four main activities of the ADC-1 are controlled by bits 7 and 8 of the command byte. The other 6 bits modify the action specified by the highest order bits. The four command groupings are:

- | | <u>Bits 8,7</u> |
|--|-----------------|
| a) Start the A/D converter on the specified channel.
Bits 1 thru 4 select the 16 analog input channels. Bit 5 selects the instrumentation amp or the onboard temperature sensor.* | 00 |
| b) Set the controlled outputs on or off.
Bits 1 to 6 control which output is set. | 01 |
| c) No operation except reset latched digital inputs.
This command allows bits 6 and 5 to determine the returned data without causing any new task to be initiated. Bits 4 thru 2 are not used. Bit 1 determines whether the latched digital inputs are reset and is only applicable when bits 5 and 6 are both 0. | 10 |
| d) Transmit a command or address on the AC powerline.
Appendix C-4 lists the AC control codes. | 11 |

Further details on the command byte may be found in Appendix C-1.

* The instrumentation amplifier can only be selected for analog inputs 1 to 8. This means bit 4 is off. When both bits 4 and 5 are on during an A/D start command, the onboard temperature sensor is selected.

3.2 Response Byte Structure

In response to each command byte sent, the ADC-1 returns a byte. The value of bits 5 and 6 in the command byte determine what information is returned. The four possible responses are:

- | | |
|--|----------|
| | Bits 6,5 |
| a) A/D low byte - All eight bits are used to carry the | 01 |
| least significant eight bits from the 12-bit A/D output. | |
| b) A/D high byte - The four least significant bits | 10 |
| contain the high order bits from the 12-bit A/D output. Bit 5 determines the sign of the voltage, bit 6 denotes an overvoltage condition, and bit 8 gives the busy/done status of the A/D converter. | |
| c) Digital input states and BSR status - bits 1 to 4 | 00 |
| carry the on/off status of digital inputs 1 to 4. Bit 5 denotes the busy/done status of the BSR transmitter. Bits 6, 7, and 8 do not hold any useful information and should be discarded. | |
| d) Garbage - the returned byte has no meaning. | 11 |

The simplest way to operate the ADC-1 is to send command bytes with bits 7 and 8 set to 00, 01, or 11 when you wish the ADC-1 to perform a task, and set to 10 when data is desired. In this way, no new activity is initiated on the ADC-1 while data is being returned. However, note that commands to activate one feature may also cause the ADC-1 to send back useful information. For example, decimal command 199 causes the BSR transmitter to send an "OFF" over the AC lines but also causes the ADC-1 to return the status of the digital inputs. Appendix C-2 contains a full list of all 256 possible commands and the resulting ADC-1 action.

It is advisable to read the response byte after sending a command byte, even if it is of no informational value. This will insure that the response byte corresponds to the command issued. Since a byte is returned for each command, failure to match responses with commands can result in errors, especially if your computer has an input buffer.

The busy/done status of the A/D converter and AC transmitter should be checked prior to issuing a subsequent command to the ADC-1 feature just activated. Bad data will result by reading the A/D before it has completed a

conversion. The symptom of not waiting for the A/D to finish is data which varies rapidly and does not seem to relate to the sensor output. Check to be sure the A/D is done by temporarily adding a wait step of about 1 second between starting the A/D and reading the high and low bytes. If the values stabilize, your problem is in that portion of your code which examines the A/D busy/done status bit.

Similarly, sending a second BSR control code to the ADC-1 before the first has finished, will result in neither command being successfully transmitted.

The advanced programmer should note that each of the four main functions of the ADC-1 can be addressed independently. While one segment is performing its task, another segment can be accessed. For example the digital inputs can be examined several times while waiting for the A/D conversion to be completed.

Program control of the ADC-1 is illustrated through the use of Microsoft BASIC 5.1, which is standard to many CP/M operating systems. The ability to use the BASIC command "OUT I,J" and the "INP(I)" function is assumed. However, PEEK and POKE can replace these commands as illustrated in chapter 2. Coding for alternate versions of BASIC where these features are not implemented or are replaced by other commands may be found in Appendix B. An example of bit manipulation by integer mathematics is included for those systems which do not support logical AND and OR functions.

Users of computers such as the Apple II, VIC-20/Commodore 64, Atari 400/800, Epson HX-20, and the IBM-PC will find subroutines specific for their computers in Appendix B.

In the following programs, the lower case comments do not need to be entered into your computer. The line numbers for each section of code were chosen so that after performing each test, you will be left with a set of subroutines (GOSUB's) which you can use in your applications programs. The subroutines were given high line numbers to keep them out of the way. Subsequent test programs will frequently use much of the same code and procedures of the preceding tests. Once a working version of a test program is in your computer, it can be SAVE'd to be used later in your custom programs.

3.3 Testing Communications

Communications with the ADC-1 are conducted through an input/output (I/O) port. Some computers allow the user to access the ports directly. Other computers, including the Apple and the Atari, require placement of data in a memory location which is then transferred out the serial port.

When addressing the I/O port directly, there are generally two numbers (sequential integers between 0 and 255) assigned to a given RS-232 serial port. These two numbers, the data port and the status port, are used by the software to reference the port and transmit data to the correct peripheral device. The port numbers can be selected by switches on some computers, while others may be firmly defined. The status port refers to a register which can be examined to determine if the byte has been completely transmitted or received. Normally, an operating system such as CP/M does this but since these programs directly access the I/O port, the user must check to make sure the transmission is done. Within the status port, there is one bit which signals when the port has received a character. This bit can be repeatedly checked until a byte has been received. The following programming examples require both the status and data port numbers and the position within the status byte of the bit which indicates transmission done.

For computers which use memory addresses for their I/O port, status and data information are placed in different locations. These locations may be accessed by using PEEK and POKE. In the following programs, substitute:

```
PEEK(dml) for INP(pn)
PEEK(sml) for INP(s)
POKE dml,j for OUT pn,j
```

where: dml - data memory address
sml - status memory address
pn - data port number
s - status port number
j - command byte (integer 0-255).

The following BASIC program is provided to test the communication of the ADC-1 with your computer or modem. The program reads the high and low bytes of the A/D without starting the conversion process. By comparing sequential readings the program will determine if RS232C connections have been made properly. If the ADC-1 and host computer baud rates are different the listed program will so indicate. Be sure to alter lines 10, 20, and 30 to reflect the port numbers on your system.

```

10  PN=6                      :REM serial port data address
20  S=7                      :REM serial status port address
30  M=2                      :REM serial input status mask
40  Y=INP(PN)                :REM clear input port of old garbage
45  FOR I=1 TO 100           :REM test enough to find problem
50  GOSUB 5020                :REM go to channel start/read subroutine
55  NEXT I                   :REM loop till I=100
60  PRINT "TEST COMPLETE - COMMUNICATIONS SUCCESSFUL "
100 GOTO 40                  :REM repeat forever

5020 OUT PN,129+32           :REM command to return A/D high byte/status
5030 GOSUB 5200               :REM wait for reply
5040 FT=CHIN                 :REM store first try at input byte
5050 OUT PN,129+32           :REM issue command to send back same byte
5052 GOSUB 5200               :REM wait for reply
5055 IF CHIN <> FT THEN GOSUB 5320 :REM if both tries not equal then error
5060 OUT PN,129+16           :REM command ADC-1 to return A/D low byte
5070 GOSUB 5200               :REM wait for reply and read byte
5080 FT=CHIN                 :REM store first try at low byte
5090 OUT PN,129+16           :REM send back second copy of low byte
5100 GOSUB 5200               :REM wait for reply and read reply
5105 IF CHIN <> FT THEN GOSUB 5320 :REM if both tries not equal then error
5110 RETURN

5200 IF(INP(S) AND M)=0 THEN GOTO 5300 :REM wait until byte is received
5210 CHIN=INP(PN)            :REM store response from ADC-1
5215 J=0                     :REM reset wait counter
5220 RETURN                  :REM return to calling program

5300 J=J+1                   :REM increment wait counter
5310 IF J<400 THEN GOTO 5200 :REM if short wait ok, else bored
5320 PRINT "BAUD RATE NOT SET RIGHT OR BAD CONNECTION" :REM problem

```

3.4 Programming the Analog to Digital Converter to Read the Onboard Temperature Sensor

A temperature sensor is provided within the ADC-1 to provide a reference temperature for use with thermocouples. The temperature sensor provides the user with the capability to make corrections to A/D readings under conditions of widely varying temperature. The sensor is an uncalibrated (+ 6°) Analog Devices AD590 which is configured to produce 1 millivolt per degree Kelvin. Thus the voltage sensed by the A/D in millivolts is equivalent to the temperature in degrees Kelvin. The AD590 is fully described in Chapter 5.

The previous test program can be altered to check the operation of the A/D with the addition of a command to start the A/D reading the onboard temperature sensor. The A/D channel read and convert subroutine begins at statement 5000. By setting CN=24, the onboard temperature sensor is selected. The resulting number returned in Z is the voltage reading across the input in units of 0.1 mv. This number can take on values from -4096 to +4096.

Note that the program code at line 5300 has been eliminated from the previous test program and the block at 5200 has been streamlined. This block will not be further changed, nor will the 5000 block.

```
10  PN=6                :REM serial port data address
20  S=7                 :REM serial status port address
30  M=2                 :REM serial input status mask
40  X=INP(PN)           :REM clear input port of old bytes
50  CN=24               :REM set channel for onboard temp sensor
60  GOSUB 5000           :REM goto channel start/read subroutine
61  REM                :REM on return will have millivolts*10
70  DK= Z/10            :REM calc temperature in Kelvin = millivolts
80  DF=(9/5*(DK-273))+32 :REM convert to degrees Fahrenheit
90  PRINT "TEMP= ",DF    :REM print temperature
100 GOTO 50              :REM repeat forever

5000 OUT PN,CN           :REM select temp sensor; start A/D conversion
5010 GOSUB 5200           :REM wait for ADC-1 response to start command
5015 DIGIN=CHIN          :REM save response byte-contains digital inputs
5020 OUT PN,129+32       :REM tell ADC-1 to return A/D high byte/status
5030 GOSUB 5200           :REM wait for reply
5040 HBYTE=CHIN          :REM save high byte from A/D
5050 IF(HBYTE AND 128)<>0 THEN 5020 :REM check status for A/D finished
5060 OUT PN,129+16       :REM command ADC-1 to return A/D low byte
5070 GOSUB 5200           :REM wait for reply and read byte
5080 LBYTE=CHIN          :REM read low byte from A/D into LBYTE
5085 HMASK=HBYTE AND 15  :REM mask for 4 high order bits from A/D
5090 Z=LBYTE+256*HMASK    :REM combine all 12 bits from A/D
5100 IF (HBYTE AND 16)=0 THEN Z=-Z :REM fix sign if negative flag set
5110 RETURN
5200 IF(INP(S) AND M)=0 THEN 5200 :REM waits for response byte to come
5210 CHIN=INP(PN)        :REM get response from ADC-1
5220 RETURN              :REM return to calling program
```

3.5 Selecting and Reading Any Analog Input Channel

The routine in the last subsection can now be used to select any analog input. Only the initial portion of the code needs to be changed in order to allow the selection of any input channel. The analog inputs are numbered on the ADC-1 terminal strip from 1 to 16, while the commands to access these inputs run from 0 to 15. Care must be exercised not to confuse the two.

```
10 PN=4                :REM serial port data address
20 S=5                 :REM serial status port address
30 M=2                 :REM serial input status mask
40 X=INP(PN)           :REM clear port of old bytes
50 FOR CN= 0 TO 15     :REM sequentially run through all channels
60 GOSUB 5000           :REM goto channel start and read subroutine
                        :REM on return will have millivolts*10
70 MV= Z/10            :REM convert to millivolts
80 PRINT CN+1,MV        :REM print channel number and value
90 NEXT CN              :REM goto next channel
100 PRINT "16 CHANNELS DONE" :REM separate loops
110 GOTO 50             :REM repeat forever
```

With the successful completion of this program, you will have a fully operational 16 channel analog to digital data acquisition system. The program for an operational data display and control program is shown in Appendix B-2.

3.6 Programming for Digital Input Operation

Each of the four digital inputs can be configured to trigger on either the opening or the closing of the line to the input. These are called normally closed (NC) and normally open (NO), respectively. Upon receiving a pulse, the input will either stay triggered until reset (latched), or will simply toggle (change state). As shipped, the digital inputs are configured as follows:

Input		
1	NC,L	trigger on opening, latched
2	NC,T	trigger on opening, toggle
3	NO,L	trigger on closing, latched
4	NO,T	trigger on closing, toggle

The user may reconfigure the inputs to any combination of these options. Section 4.3 describes how to rewire a header block for your specific needs.

Although the ADC-1 does not include an accumulator, counts of events can be obtained by monitoring an input and incrementing a counter within the user's program. If communications with the ADC-1 are set at 9600 baud, events occurring up to 400 times a second can be counted.

The following program monitors pulses and contact closures/openings for each of the digital inputs. Note that the wait loop at line number 5200 from the last program is used again to check for the arrival of incoming bytes.

```

10  PN=6                :REM serial port data address
20  S=7                 :REM serial status port address
30  M=2                 :REM serial input status mask
40  X=INP(PN)           :REM make sure input buffer is clear
50  GOSUB 6000           :REM go read the digital inputs
60  FOR I=1 TO 4        :REM check each input
70  DTEST=2^(I-1)       :REM create mask for input bit to be tested
80  IF DTEST AND D THEN PRINT "ON",ELSE PRINT "OFF", :REM check each input
90  NEXT                :REM go test next input
100 PRINT               :REM skip a line
110 GOTO 50             :REM start over, loop forever

5200 IF(INP(S) AND M)=0 THEN 5200      :REM waits for response byte
5210 CHIN=INP(PN)                     :REM get response from ADC-1
5220 RETURN                           :REM return to calling program

6000 OUT PN,129                       :REM read digital inputs without resetting
6010 GOSUB 5200                       :REM wait for communications to be done
6030 D = CHIN AND 15                  :REM mask for wanted input bits
6040 OUT PN,128                       :REM reset latched inputs
6050 GOSUB 5200                       :REM wait, read byte to clear port
6060 RETURN                           :REM

```

As you run this program, the operation of the four digital input configurations can be demonstrated. Use a short piece of wire to momentarily connect the positive and negative sides of each digital input on the ADC-1 terminal strip. The rolling display will indicate when an input has changed state. You should be able to change columns 2 and 4 from ON to OFF and back by momentarily shorting inputs 2 and 4. This illustrates the toggle feature.

Try this with inputs 1 and 3. You should see the display indicate ON but then rapidly alter to OFF. These input were triggered, but rapidly reset and ready to be triggered again. You may want to add a delay to make this more visible. Adding 105 FOR J=1 TO 1000: NEXT should accomplish the delay.

3.7 Program Control of AC Line Carrier Modules

Each BSR module has two dials or rotary switches which are used to assign a house code ("A" - "P") and a unit code (1 - 16). The house code is designed to allow nearby houses or offices to possess control systems without interfering with each other. The unit code allows up to 16 modules to be addressed independently. To select a house code for the ADC-1, refer to Appendix C-4 for the required switch setting. Alter the settings of the bank of 8 switches located closest to the edge on the bottom of the ADC-1. As shipped, the ADC-1 is set to select house codes "A" and "I".

Each setting of the dip switches selects a pair of house codes for a total of 32 separate unit codes. The full list of the 32 possible unit address codes is shown in Appendix C-4. Thus, if the switches are set for house (A,I), and a "250" is sent to the ADC-1 via the command byte, the BSR module with the setting "I 8" will respond.

Once a unit code has been transmitted, that BSR unit can be repeatedly turned on/off by sending command codes without repeating the unit code. To control a different BSR module, simply designate a new unit code. More than one module can be set with the same unit and house codes. In this case, all units with the same code will be controlled simultaneously.

In some circumstances, the address and control codes should be sent twice. If electrical noise on the AC wiring occurs at the same instant that a code is being transmitted, the code may not be received properly by the BSR module.

* The following program will request a BSR module number, and then allow the user to turn that module on or off, or cause a lamp plugged into the BSR module to "count" to five. The "counting" demonstrates the maximum speed at which the BSR codes can be transmitted. Note that line 7020 which checks the BSR busy/done status, also causes the ADC-1 to return the status of the digital inputs. For some applications, such as for a security system, the user may desire to continually evaluate these input states.


```

5   DIM DEVICE(16)           :REM array for device codes
10  PN=6                     :REM specify data port
20  S=7                      :REM serial status port address
30  M=2                      :REM serial input status mask
40  X=INP(PN)                :REM clear port of old bytes
50  FOR K=1 TO 16            :REM read in 16 unit address codes
60  READ DEVICE(K)           :REM get codes from DATA line 7900
70  NEXT K                   :REM loop till 16 codes
80  INPUT "WHAT BSR MODULE NUMBER SHOULD I CONTROL? ",BM
90  INPUT "SHOULD I TURN IT ON (1), OFF(2), OR MAKE IT COUNT (3)",ACT
100 UC=DEVICE(BM)            :REM select device address code
110 GOSUB 7000               :REM send code and wait for ADC-1 done
120 IF ACT=1 THEN UC=197 :GOSUB 7000 :REM if 1 then send an ON
130 IF ACT=2 THEN UC=199 :GOSUB 7000 :REM if 2 then send an OFF
140 IF ACT=3 THEN GOSUB 180 :REM if 3 then make lights flash
150 GOTO 80                  :REM go ask for new module number

180 UC=199                  :REM make sure light off before counting
190 GOSUB 7000              :REM send OFF to unit
200 FOR J= 1 TO 5           :REM BSR unit will count to five
210 FOR I=1 TO J            :REM loop to send each "count" sequence
220 UC=197                  :REM turn unit ON
230 GOSUB 7000              :REM send BSR code, wait till done
240 UC=199                  :REM turn unit OFF
250 GOSUB 7000              :REM send and wait
260 NEXT I                  :REM loop to finish this number
270 FOR K=1 TO 1000:NEXT    :REM wait before sending next flashes
280 NEXT J                  :REM goto next number of flashes
290 RETURN

5200 IF(INP(S) AND M)=0 THEN 5200 :REM waits for response byte to come
5210 CHIN=INP(PN)           :REM get response from ADC-1
5220 RETURN                 :REM return to calling program

7000 OUT PN,UC              :REM send unit code to ADC-1
7010 GOSUB 5200             :REM get ADC-1 response
7020 OUT PN,129             :REM ask for BSR busy/done status
7030 GOSUB 5200             :REM wait for answer
7040 IF (CHIN AND 16)<>0 THEN 7020 :REM if still busy, ask again
7050 RETURN                 :REM done, return to main program

7900 DATA 204,220,196,212,194,210,202,218,206,222,198,214,192,208,200,216
7901 REM                    module addresses for units 1 to 16

```


3.8 Programming the Controlled Outputs

The following program will turn on or off any of the six controlled outputs of the ADC-1. The program requests the number of the controlled output as labeled on the ADC-1 terminal strip. The location of the bit in the command instruction designates which output, as labeled on the connector, is to be controlled. Thus the LSB in the command instruction is assigned to controlled output #1. The 6th bit then corresponds to output #6. The outputs are numbered on the terminal strip with #1 located nearest the bottom edge.

The statement on line number 8000 produces the command to alter the state of the outputs by checking the array DS and adding in the proper bit values. Initially in the program, all six outputs are set OFF. You can then change them one at a time and check the results with a voltmeter. When the output is on, it should read +5 volts.

```
5   DIM DS(6)                :REM array for controlled output states
10  PN=6                      :REM specify data port
20  S=7                       :REM serial status port address
30  M=2                       :REM serial input status mask
40  X=INP(PN)                 :REM clear port of old bytes
50  FOR I=1 TO 6              :REM initialize the outputs to OFF
60  DS(I)=0                   :REM zero is off, one is on
70  NEXT I                    :REM end loop to set all outputs OFF
80  GOSUB 8000                 :REM output command to set all outputs OFF
90  INPUT "CONTROLLED OUTPUT NUMBER (1-6): ",J
                               :REM get new output number
100 INPUT "TYPE 1 FOR ON AND 0 FOR OFF : ",DS(J)
                               :REM put new state in array
110 GOSUB 8000                 :REM send new output status
120 PRINT "DONE"              :REM when byte sent and done, type done
130 GOTO 90                   :REM ask for new channel.
                               changes remain in array

5200 IF(INP(S) AND M)=0 THEN 5200 :REM waits for response byte to come
5210 CHIN=INP(PN)              :REM get response from ADC-1
5220 RETURN                   :REM return to calling program

8000 DOUT=DS(1)+2*DS(2)+4*DS(3)+8*DS(4)+16*DS(5)+32*DS(6)+64
8010 OUT PN,DOUT               :REM send output control byte set in DOUT
8020 GOSUB 5200                :REM wait for communications to be done
8050 RETURN                   :REM return to calling program
```

You have now tested each feature of the ADC-1. The subroutines listed in lines 5000 to 8000 should form the nucleus of your custom applications program. Good luck and enjoy your computer's new capabilities.

4.0 Details

This chapter presents detailed information on the operation and options available in the hardware of the ADC-1. Instructions on the repair of some common user-induced problems are included.

Before any repairs or modifications can be started, the printed circuit board must be removed from the aluminum enclosure. After removing the four rubber feet, the bottom plate (with the printed circuit board attached) can now be eased from the enclosure. Let the analog inputs terminal strip drop from the enclosure slots first, then slide the board and base plate from the enclosure. The terminal strips may stick or bind a small amount, but can be gently adjusted to allow enclosure removal or re-installation.

4.1 Communications

The ADC-1 uses an Intersil 6402 CMOS UART capable of communication rates of 250k baud. The baud rate set by SW 2 on the ADC-1 determines both the maximum cycle rate of the TTL controlled outputs and the maximum counting frequency from the digital inputs. Both the A/D and AC line transmitter are frequency limited, so that communication rates greater than 1200 baud do not appreciably increase throughput for these functions.

Since most RS-232 serial ports and modems can select either 1 or 2 stop bits, the ADC-1 has been hard wired for 2 stop bits. If you are unable to configure your system for two stop bits, the ADC-1, as shipped, may function using one stop bit. However, it is possible to alter the printed circuit board of the ADC-1 to specify one stop bit. Cut the trace on the component side of the board between pins 35 and 36 and between pins 36 and 37 on the 6402 UART (U28). Insert a jumper connecting pins 35 and 37. Connect a jumper between pin 3 (ground) and pin 36 on the 6402 UART (U28) to select one stop bit.

Communication to the UART may be performed using TTL levels (+ 5 volts) allowing direct operation with the VIC-20 and ATARI 400/800. TTL input to the UART is on pin 21 of the ADC-1 DB-25 connector. The UART output is tied to pin 22 of the DB-25. The CMOS UART has limited output driving capability and thus should drive only a single input over a line length not to exceed six feet.

When connecting directly to the UART TTL level inputs (DB-25 pins 21 and 22) care must be taken to assure proper connection of transmitted and received data. If the ADC-1 UART output is connected to the computer output, damage may result to the UART (U28). The 6402 UART may be replaced if damaged by removing it from the 40 pin IC socket and inserting another UART.

4.2 Analog to Digital Converter

observed analog input range

no instr. amp.

instr. amp

$\pm 0.4096 \text{ V}$

$\pm 0.04096 \text{ V}$

10-10-84 10:30 AM

selected $\rightarrow$ The A/D converter (U29) is an integrating dual slope A/D with 12 bits of resolution plus sign. As shipped the A/D conversion rate is 7 hz set by SW2 switch 8. Switch 7 on SW2 sets the A/D clock frequency to yield a conversion rate of 14 hz. At 14 hz. the integrating output from the A/D is half the optimal range and may show slightly greater noise at low level signals. For proper operation either switch 7 or 8 must be on for SW2. Improper operation will result if both switches are off or on.

The conversion rate may be altered to a maximum of 30 hz while still retaining auto-zero operation. An explanation of the speed capabilities of the A/D is included in the Intersil 7109 technical bulletin (Appendix D). Both the integrating resistor R35 and the integrating capacitor C15 should be determined from the equations in the technical bulletin.

Recent tests have resulted in successful operation of the ADC-1 7109 to speeds of 100 Hz. However, at this speed the auto-zero feature of the 7109 does not function. Under normal low speed operation, the digital number resulting from a shorted analog input will be zero. At speeds greater than 30 Hz., each 7109 will settle to a non-zero digital value unique to that chip when shorted. After this number has been determined, it can be subtracted within the software. Noise levels are ± 4 bits at this speed.

The ADC-1 A/D is shipped with an input range of ± 0.4096 volts and a resolution of 100 microvolts. The input range can be modified to accept an input range of ± 2.0 volts yielding a resolution of 0.5 millivolts. To make major changes in the voltage input range it is necessary to alter the integrating resistor R35, the integrating capacitor C15, the auto zero capacitor C14, and the

- Note: 1. "Wiggle" IC's on circuit board
2. jumper input - TO unit ground (2 CND screws available)
See this page and pages 35-36

reference trimpot R1 as specified in the Intersil technical bulletin in appendix D-1. Small changes in the A/D input range and resolution may be made by adjusting R1 so that the voltage across pins 36 and 39 of the A/D (U29) is one half the desired input range. The ADC-1 is shipped with R1 adjusted to provide 0.2048 volts on the A/D reference inputs (pins 36 and 39 of U29), thus determining an input voltage range of ± 0.4096 volts. Both reference and analog inputs are wired to operate in the differential mode.

The ADC-1 is equipped with an Analog Devices AD589K voltage reference which provides an external, temperature compensated, reference voltage for the A/D. The AD589K has a temperature coefficient of 50 ppm/°C. It can be replaced by a more costly AD589M to provide temperature stability as low as 10 ppm/°C.

Conversion commands to the A/D are routed through a 4528 CMOS monostable multivibrator to delay starting the A/D. The length of the delay is 0.01 seconds to allow settling of the analog inputs prior to starting the A/D. The delay time chosen is considered to be a minimum time delay. This delay length functions best with sensors of low output impedance. Differential signals derived from sensors of high output impedance may not settle to their actual analog level with the length of delay implemented. The delay time may be increased by increasing the value of either R11 or C7. R11 should not be increased beyond 100K ohms. The amount of increase in either of these components can be determined experimentally or from the timing diagrams of a 4528 monostable multivibrator contained in most CMOS data manuals. An alternate method of increasing the delay is within the software. Program the A/D to make two conversions for each channel sampled, selecting the second conversion to characterize the voltage. This procedure should be adopted when using the high gain instrumentation amplifier. The amplifier is rated for 10 hz operation. However when switching between analog channels of differing input levels the amplifier may not reach the final level in a single A/D conversion.

For applications where the common mode input voltage is greater than the ADC-1 supply voltage, it is desirable to set the input low of the A/D and the sensor input to ground. To do this, a location marked B on the component side of the PC board next to pin 23 of U29 may be jumpered to a ground pad located immediately below C13.

*

Analog Inputs

Inputs to the 16 input analog switches must be limited to levels within the supply voltage of the ADC-1 board, e.g., +5 volts. Inputs which exceed this range are likely to damage the CMOS 4051 analog multiplexors. If a 4051 (U17 to U20) analog switch is damaged, an IC socket allows easy replacement. A four channel 4052 (U27) analog multiplexor controls final selection of inputs to the A/D. This set of 4 switches connects the A/D to either of two groups of eight analog inputs, to an output of an optional instrumentation amplifier hardwired to the first 8 analog inputs, or to a temperature sensor mounted on the ADC-1.

Instrumentation Amplifier

The optional instrumentation amplifier (U34) is an ICL7605 CMOS commutating auto-zero amplifier. Use of the instrumentation amplifier with the ADC-1 permits the measurement of microvolt inputs from low signal level sensors such as thermocouples. For most thermocouple types, the ADC-1 can resolve 0.2°C over a 800°C range. The amplifier is connected to the first eight analog channels. As shipped the amplifier provides 10 microvolt resolution for a full scale range of 40 mv. Changing a single resistor to the left of location marked D on the board will allow gains from 1 to 100 to be defined by the user. Gains greater than 100 resulting in resolution better than 1 microvolt are unreliable due to small stray currents arising from the analog multiplexors.

The gain of the amplifier can be changed by changing resistors R52 and R51 along with the trimpot R50. The equation for the gain using the named resistors is:

$$\text{Gain} = (R50 + R51 + R52) / (R51)$$

The value of the trimpot R50 should be selected to be 10 percent of R52 to minimize effects of temperature with the greater temperature coefficients of trimpots over metal film resistors. Both R51 and R52 should be metal film or wire wound resistors with temperature coefficients less than 50 ppm. The trimpot R50 should be selected with a temperature coefficient less than 50 ppm/ °C.

4.3 Digital Inputs

The four digital inputs of the ADC-1 trigger when they sense a change in logic level. They do not sense a constant logic level. CMOS 4013 flip flops (U12 and U13) are used to monitor the status of the digital inputs. The ability to sense rapidly occurring digital events is limited by the baud rate. At 9600 baud, it is feasible to monitor inputs operating at 400 hz.

As shipped, the ADC-1 is configured with two normally open (inputs #3 and #4) and two normally closed (#1 and #2) digital inputs. Since 4013's trigger on low to high voltage transitions, for the normally closed inputs, the 4013 inputs are pulled low when a closed switch is across the plus and minus ADC-1 input. If this short is broken, thus opening a normally closed circuit, the 4013 triggers. The plus side of digital inputs 1 and 2 are each connected to the positive supply voltage with a 20k resistor (R8 and R9) to provide normally closed operation.

For normally open inputs, the input to the 4013 is low until a closed circuit across the ADC-1 input terminals forces the 4013 input line high. In this configuration, the ADC-1 digital input terminal labeled minus is strapped via a 600 ohm resistor to provide +5V. The plus side of digital inputs #3 and #4 are each connected to ground with a 20k resistor (R6 and R7) to provide for normally open operation. This allows contact closures at the inputs to produce a pulse which is sensed by a 4013 CMOS flip-flop. The size of the resistor is sufficiently high so that the inputs can be driven directly from other CMOS or TTL outputs provided the logic levels are TTL compatible.

Care should be taken for switches mounted outside, since moisture may cause the resistance in wiring connected to the switches to drop well below 20k, thus causing false events on the digital inputs. The inputs can be made less sensitive by lowering the resistors R6, R7, R8, and R9 to 5k at an increased load to the power supply. Additionally, the resistors can be changed allowing any combination of normally open or normally closed circuits to be monitored on the four digital inputs.

Make sure when connecting a switch to the digital inputs, that you connect to the two opposing inputs on the terminal strip. Also, it is possible to damage the 4013's (U12 and U13) if the digital inputs are driven by external logic levels exceeding the 5 volt supply voltage. These IC's are socketed for easy replacement if damaged.

The digital input header shown on the component diagram allows each digital input to be configured for latched or pulsed operation. Latched digital inputs once set do not change state until reset by a command from the connected computer or modem. Pulsed inputs on the other hand toggle in response to events on the input line. In this mode, the computer must monitor the line at twice the highest expected frequency of events to correctly determine the integral count, average rate, etc.

The following table provides pin assignments necessary to configure the digital inputs header for pulsed or latched mode. Each digital input flip-flop has an output, data, and reset pin connected to the digital input header. To configure a given digital input for latched operation connect the data pin to +5, and the specified reset to the reset control. To set a given digital input to count pulses (toggle), connect the channel output to the equivalent numbered data pin. Next connect the reset of that digital channel to ground. The position of pin one (the beveled edge) of the digital input IC header is as indicated on the component diagram.

Digital Input Header Pin Assignments

<u>Pin No.</u>	<u>Signal Description</u>
1	data for digital channel #4
2	data for digital channel #3
3	data for digital channel #2
4	data for digital channel #1
5	reset for digital channel #4
6	reset for digital channel #1
7	reset for digital channel #3
8	reset for digital channel #2
9	ground
10	+5 volts
11	reset control
12	no connection
13	output for digital channel #1
14	output for digital channel #2
15	output for digital channel #3
16	output for digital channel #4

For contact switches with excessive switching noise, adding a small capacitor will filter the high frequencies so that reliable operation is obtained. The capacitor should be placed between the positive and negative side of the input on the ADC-1 terminal strip.

4.4 BSR Control

The ADC-1 AC line carrier control allows the connected computer to control 32 separate BSR devices. Selection of a pair of house codes, listed in Table C-3, is provided through SW1 on the ADC-1 board allowing the 256 possible BSR devices to be addressed. The non-sequential nature of the house code pairs and of the unit address codes is a result of BSR's digital coding pattern.

The signals which control the BSR modules are transmitted on the AC wiring via a serial code which is impressed on a 120 kHz carrier wave. This transmitted code is synchronized with each module by using the zero crossings of 60 hz AC voltage. Each bit of the code represents a + 5 volt modulation of the 120 kHz for 1 millisecond. Each nine bit message is transmitted in true and inverse form and is followed by an end code. The total message time takes 0.4 seconds. To accomodate multiphase wiring the code is repeated three times at phases equally spaced over the 120 hz half cycle.

Warning!

The BSR signal transmitter is housed in a small plastic box connected to the main body of the ADC-1 via a flat ribbon cable. This was done in order to separate any possible AC interference from the sensitive A/D section. Hazardous AC voltages are present on several pins within the small enclosure. Do not open the enclosure while the unit is plugged into an AC outlet.

4.5 Controlled Outputs

The six controlled outputs provide buffered TTL level outputs using 2N2222A transistors. Pin 14 on the DB-25 of the ADC-1 is connected to the collectors of the transistors (Q1 to Q6) which correspond to the six controlled outputs. A separate supply may be used for current draws exceeding a total of 5 millamps for the six controlled outputs. The unit comes shipped with a 600 ohm resistor between pin 14 and 18 of the ADC-1 DB-25 to limit the total current consumption of the six outputs. This resistor can be disconnected and a separate supply connected for larger current outputs. Operation of each transistor of the controlled outputs should be limited to less than 50 milliamps at 5 volts to minimize ground noise levels for the A/D, and to prevent damage to the 2N2222 transistors.

Should a controlled output cease to function it is most likely a failure of the transistor associated with the controlled output. Replace the transistor in line with the output label on the ADC-1 controlled output terminal strip. If the controlled outputs are used with small motors or relays they should be diode protected.

The controlled outputs were designed to drive solid state AC relays made by a variety of manufacturers including International Rectifier ("Crydom") and Motorola. Individual controlled outputs may be connected to optoisolators to provide greater isolation for controlling small DC motors, lights or relays.

4.6 Applications Using The ADC-1 Board Select

Connection of the ADC-1 to computers such as the ATARI 400/800 require a board select control because multiple IO devices share the same serial input line. When pin 8 on the DB-25 of ADC-1 is grounded, the ADC-1 is deselected, and is prevented from accepting further commands sent on the serial line. If no connection is made to pin 8, then that line is held high internally and the ADC-1 is operational. This feature allows the use of more than one ADC-1 with the same computer I/O port via a hardwire select/deselect line.

4.7 Powering The ADC-1 From Batteries

The ADC-1 requires 20ma at 5 Volts to operate and will function well with a battery power supply. Any battery with a voltage output of 4.5 to 24 volts may be used.

The least expensive and simplest battery supply is three "D" cells wired in series which provides 5.0-4.4 volts and is connected directly to the ADC-1 +5 volt power input on pin 18 of the ADC-1 DB-25. In a test, three Eveready Super Heavy Duty "D" cell batteries (\$.53 each) powered the ADC-1, 7 AD-590 temperature sensors, and 3 resistive networks (measuring soil moisture, water level, and battery voltage) for over 190 hours(>1 week of continuous operation).

A six volt lantern battery can be used and connected to the +5 volt power input (pin 18) if two inexpensive diodes (such as a 1N4148) are connected in series between the battery and the ADC-1. Each diode drops the voltage about 0.5 volt and consumes less current than a resistor. The power from a 12 volt automotive battery should be input to the ADC-1 through pin 9 of the DB-25 connector. In this case the voltage is internally lowered to +5 V by a zener diode in the ADC-1 and the current consumed is greater than for a +5 volt supply. Please contact Remote Measurement Systems for suggestions to reduce the ADC-1 requirements for applications with stringent power supply restrictions.

5.0 Analog and Digital Sensor Descriptions

The sensors described in this chapter are examples of transducers in common use, and are not intended to be a comprehensive list. The particular sensors discussed are known to function well with the ADC-1. However, the ADC-1 is capable of functioning superbly with a wide array of transducer and sensors. As users report the successful implementation of additional sensors, Remote Measurement Systems will expand this chapter.

5.1 Installation and Operation

The ADC-1 is equipped with a pair of terminals for each of 16 analog inputs and 4 digital inputs. These connectors are located on the top lefthand edge of the ADC-1. Attachment of sensor leads is easily accomplished with the aid of a screwdriver to close the vise-like terminal. The analog terminal strips are labeled "+" and "-" to denote the polarity sensed by the A/D converter. For example, a minus reading from the A/D means that the sensor lead attached to the "-" terminal is at a higher potential than the lead attached to the "+" terminal.

Special types of wire are generally not required to interface sensors (other than thermocouples) with the ADC-1. In electrically noisy environments it may be advisable to use shielded cable. Long unshielded sensor leads may act as antennas and pickup unwanted signals, especially with high impedance sensors (>10K ohms) such as pH probes and CdS photoresistive cells.

In checking the installation of your sensors, you may notice that the values returned by the ADC-1 appear to be very noisy and variable. A likely cause of this condition is that the ground of your sensor is floating relative to the ADC-1 ground. This can be checked by connecting a short length of wire between the negative side of the analog input and either of the two terminals labeled GND. If the unwanted noise is reduced, then you may want to leave the jumper in place. Alternately, for a more permanent installation, pads are provided on the printed circuit board for the placement of ground jumpers. Ground is available on pads between inputs 4 and 5 and between 11 and 12. The sensor low (ground) must be within +3 volts of the ADC-1 ground. Common mode voltages outside this range will cause the ADC-1 to produce noisy readings.

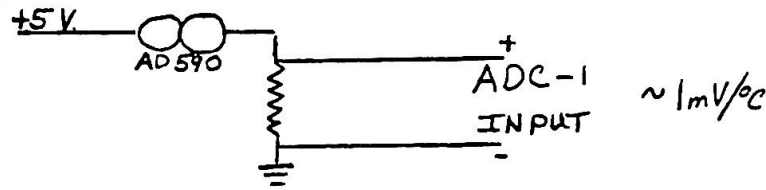
*

* If grounding the input does not reduce the noise to acceptable levels, you may need to filter the input. By placing a small capacitor (10-50 uf) across the input, high frequency spikes will be eliminated. As larger capacitors are used, lower frequencies are filtered. For some applications, you may need to refer to a book on filtering and signal conditioning.

5.2 Temperature Sensors

Solid State Sensors

The Analog Devices AD590 is a two terminal integrated circuit transducer producing an output current proportional to absolute temperature. For supply voltages between +4V and +30V, the device acts as a high impedance, constant current regulator passing $1\mu\text{A}/^\circ\text{C}$. The range and sensitivity of the sensor is determined by the value of the load resistor which converts current to voltage. Connection to the ADC-1 is possible with the circuit:



In this case, the load resistor is 1K ohm, yielding 1 millivolt per degree centigrade. Since the ADC-1 senses voltages of 0.1 millivolt, temperature variations of 0.1 °C can be measured over a range of -55°C to 135°C.

Individual AD590 sensors are uncalibrated. An accuracy of .1°C is obtainable with a gain and offset correction determined empirically by the user. The correction may be physically performed with a variable resistor or may be easily calculated in the software. The largest of the two errors, the offset, is a calibration error and is constant over the temperature range. To correct this error, a constant offset, determined near the center of the desired measurement range, may be added or subtracted from all temperatures. After this correction is made, the remaining error is non-linear and varies with temperature. This gain error may also be corrected in the software but requires additional calibration points.

The AD590 is graded according to the magnitude of the errors. The J grade offset error is $+5^{\circ}\text{C}$ which when corrected, leaves a gain error of $+1.5^{\circ}\text{C}$. The M grade offset error is $+0.5^{\circ}\text{C}$ leaving a non-linear gain error of $+0.2^{\circ}\text{C}$. However, the M grade is about 10 times more expensive than the J grade. The J grade was selected for the onboard temperature sensor as it offers a reasonable cost/accuracy compromise. If you intend to use the ADC-1 onboard temperature sensor for measurements, or for thermocouple junction compensation, be sure to determine the offset error. Error compensation and grade accuracy are discussed in both the Analog Devices technical data sheet and in their application note for the AD590.

Thermocouples

Temperature measurements are possible with thermocouples when the optional instrumentation amplifier is installed in the ADC-1. The use of thermocouples allows temperatures to be determined over a wide range of values. When wires constructed of different metals are joined to form a circuit, if one of the junctions is at a higher temperature, a current will flow. By placing one junction at a standard temperature, the other junction can be used to determine the unknown temperature by measuring the emf generated. The ADC-1 can be used to sample this emf(voltage).

The ease in making thermocouple junctions, combined with the standard response of each device provides an advantage over solid state devices such as the AD590 which is limited in range and must be individually calibrated. Copper-constantan (type T) thermocouples are most commonly used for applications requiring temperature measurements near ambient temperatures. However type T thermocouples are applicable over ranges from -250°C to 400°C with an average output near 40 microvolts/ $^{\circ}\text{C}$. Although thermocouple responses are well defined physically, they are non-linear and, depending on the desired accuracy and range, may require the use of polynomial equations to convert from the measured microvolt output to degrees.

For temperatures greater than 400°C , either Chromel-alumel (type J) or iron-constantan (type K) thermocouple junctions may be used, providing the capability to measure temperatures up to 1372°C . To span the full range of temperature from 0° to 1372° for type J thermocouples the National Bureau of Standards uses an eighth-degree polynomial. Note the ADC-1 can provide a 800°C range in temperature measurement with $.2^{\circ}\text{C}$ resolution. The accuracy depends on calibration procedures of the onboard temperature reference.

Measurement of temperatures using thermocouples requires a reference junction within the circuit. Generally this will be the temperature of the junction near the measuring instrument, in this case the ADC-1. The thermocouple wire can be run directly from the sample junction to the ADC-1. The remote temperature will only be determined to the accuracy of the known absolute temperature reference. Such references can take the form of constant temperature baths, or solid state reference devices. The ADC-1 uses an AD590 temperature sensor mounted to the PC board that will provide an absolute temperature reference after being calibrated. The accuracy of the onboard temperature sensor will provide $\pm 0.2^{\circ}\text{C}$ calibration if the terminal blocks are at the same temperature as the AD590. Such an assumption is valid if the ADC-1 is placed in a location of uniform or slow temperature change. Large reference errors will result if the ADC-1 is exposed to direct sunlight or wind currents characterized by large temperature variations.

The thermocouple equation and coefficients which are needed to convert the microvolt readings to temperature will depend on the materials used. The reader is referred to a text such as Fritchen and Gay's Environmental Instrumentation for details.

5.3 Wind Speed Sensors

A suggested wind speed sensor is a rotating cup anemometer manufactured by Tradewind Instruments. The anemometer mounts on standard 1/2 inch pipe and is made of black lexan. Through the use of a small generator, it produces a millivolt output which is measured directly by the ADC-1. The output of the generator was calibrated in a wind tunnel by Trade Wind Instruments. The response with wind speed is linear (.34 MPH/mv.) and is calculated by the simple equation:

$$\text{Wind Speed (MPH)} = 0.03375 * \text{DN}$$

where DN is the digital number returned by the A/D converter in units of 0.1 mV,

Wind speeds from 0 to 100 MPH. can be measured although accuracy diminishes below 2 MPH. The two leads from the anemometer are connected directly into an analog input terminal pair (+ and -).

Another type of anemometer produces pulses in proportion to the wind speed. The ADC-1 can be used to count these pulses, but to convert to wind speed requires a measurement of the time interval.

5.4 Electrical Energy Consumption Sensor

We have located a versatile and easy to use sensor to measure AC current. The Amprobe AC Current transducer A603 has four current ranges(0-5/25/100/250 Amps AC) and produces a secondary current of 0-260uAmp DC which when dropped through a resistor, provides an ideal input to the ADC-1. Though not a true power (watt) sensor, since it does not measure current and voltage simultaneously, it functions well in most home energy management applications. An estimate of the power may be calculated by multiplying the AC current measured times the voltage. For resistive consumption of electricity, such as in heaters, lights and stoves, the power thus calculated should be reasonably accurate. The largest errors will occur in situations where large inductive motors are employed.

Installation is very simple as the transducer clamp opens to allow placing around one leg of the AC conductor and then springs shut. The AC conductor does not need to be disconnected. For smaller appliances, an Amprobe model A-47L adapter, is available which connects between the appliance and outlet and allows easy clamp-on measurement. When used in combination with the ADC-1, the probe is able to resolve power variations as small as 10 watts over a range of 28000 watts. The A603 may be connected to the ADC-1 inputs in parallel with a 400 ohm load resistor, but has, in our tests, required the use of a 100+uF capacitor to reduce the noise on the A603 output. The datasheet in Appendix D contains more details, as well as a graph to convert the sensed output to AC amps. This slightly non-linear curve can be approximated by several straight line segments and included in your software enabling your computer to monitor and display electrical energy consumption directly.

5.5 Light Level Sensors

Photovoltaic and photoresistive sensors are available from a variety of electronic distributors. The inexpensive photovoltaic cells have generally proved to be simpler to install and function extremely well for home or office applications where accuracy is not required. Some photovoltaics may be connected directly across the analog inputs, while others require a voltage dropping resistor.

The sensitivity and resolution of the ADC-1 permit the use of a \$5 photovoltaic cell as a motion detector. The ADC-1 can easily measure the changing light levels as a person moves about in the ambient room light. An alarm condition can be set should the light levels vary by more than a predetermined amount in a short period of time. A second sensor monitoring outside illumination can be used prevent false triggers on fast moving clouds.

The ADC-1 can be used with more costly precision radiometers to accurately measure solar radiation. Many varieties of radiometers and pyranometers exist with outputs in the range of 2 to 16 $\mu\text{V/W/M}^2$. With the optional instrumentation amplifier, the ADC-1 can measure direct sunlight (1370 W/M^2) with a resolution of about 1 W/M^2 , or to 0.1%. We again refer to books such as Environmental Instrumentation (Fritchen and Gay, 1979) for complete descriptions of this instrumentation.

5.6 Optical Encoders

An optical encoder consists of an opaque disc with holes or reflective pads around the circumference. The disc is coupled to a light source and photo-sensitive sensor so that as the disc is rotated the holes pass the sensor generating a pulse for each hole passed. Such encoders, when coupled to rotating shafts, can provide a measure of the amount of shaft rotation. Rotation speeds can be determined by counting the number of pulses in a known time interval.

An encoder connected to a wheel can be rolled along a surface to measure distance. Optionally, the surface can be moved and the wheel held stationary. Optical encoders are useful in computer process control applications for both linear or angular positioning, and for controlling linear and angular velocities. For example, it is

possible to use the ADC-1 to determine the position of a motor driven solar panel and to control its movement to provide optimal positioning. An informative article entitled "Designing with Optical Shaft Encoders" recently appeared in Robotics Age magazine (May/June 1983, pg 27-34).

When applied with the ADC-1 the output pulse rate of encoders must be limited to 400 pulses per second. At this rate the ADC-1 must operate at 9600 baud and with a program able to sample at 1000 hz. Microsoft's BASIC interpreter is too slow for monitoring such rates.

5.7 Security System Sensors

The use of the ADC-1 to monitor security sensors allows the user flexibility not available with conventional security system controllers. The analog inputs can be employed to measure changing light or sound levels, while the digital inputs monitor door or window switches, floor pressure contacts, or other commercial detectors. Through programming, the user can set conditions and responses which will make a home or office appear to be occupied and thus discourage intrusion.

For example, a vibration sensor on the bedroom window can be monitored to sense an attempted opening. If an attempted break-in is sensed during the night, house lights are turned on in a sequence which appears as if someone is walking toward the intrusion. During the day, a loud radio could be activated. Should the intruder persist and actually open the window, a siren can alert the neighborhood that an attempted burglary is in progress. Hopefully, the rapid activation of interior and exterior lighting has scared the burglar into leaving before damaging the window.

Magnetic Switches

These switches are sold in either normally open or normally closed configurations in which a magnet is employed to maintain the switch condition. As soon as the magnet moves, the switch changes state, and the ADC-1 digital inputs can detect the change. These sensors can be chained in series (NC) or parallel (NO) loops to permit more than four switches to be monitored by the ADC-1. With the use of a small external power source (even a battery), even the ADC-1 analog inputs can be used to monitor switches. By checking the value of the sensed voltage, a change in circuit condition can be monitored.

Vibration Sensor

Litton Poly-Scientific manufactures a small one inch square vibration sensor (AC2537). Inside the plastic housing, a small ring sits balanced, making momentary contact when shaken. The pulses quickly trigger the ADC-1 digital inputs.

Floor Mat Pressure Sensor

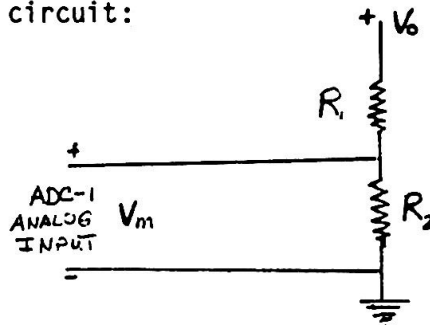
Flat strips of copper, held apart by small pieces of foam, are packaged in rolls of plastic about 8" apart. When placed under a rug and compressed by a person's weight, the strips make contact and trigger a digital input of the ADC-1. These pressure mats are available from most security system suppliers.

Infrared and Ultrasonic Detectors

The commercial versions currently available are extremely expensive. Remote Measurement Systems is developing simpler, less expensive infrared and ultrasonic sensors for use with the ADC-1.

5.8 DC Voltage Scaling

Occasionally you may want to use the ADC-1 to monitor DC voltages greater than the normal 0 - + 0.4 volt range of the ADC-1 analog inputs. These larger voltages may be reduced to the ADC-1 input range with the following simple two resistor circuit:



where V_0 is the voltage to be measured, and
 V_m is the voltage digitized by the ADC-1.

Since the current passing through each resistor is the same, by Ohm's Law,

$$\frac{V_o}{R_1 + R_2} = \frac{V_m}{R_2}$$

Therefore, the voltage you want to measure is given by:

$$V_o = \left(\frac{R_1 + R_2}{R_2} \right) V_m$$

where V_m is the voltage measured by the ADC-1 in volts. Remember that the digital number returned by the ADC-1 is in units of 0.1 mv and must be divided by 10000 to get volts.

In order for this circuit to function, the total of R_1 and R_2 must be much greater than the internal resistance of the voltage supply. In addition, the voltage source (sensor, battery, etc.) must be able to supply the current drawn by the resistors. It is therefore an excellent idea to choose the values of R_1 and R_2 to be large so as to minimize the current drawn. The total of R_1 and R_2 should not exceed about 1 Meg ohm.

For example, to monitor the status of your power supply or battery, you may want to measure voltages of 5 to 25 volts. A possible choice of values would be $R_1 = 70K$ ohm and $R_2 = 1K$ ohm. In this case $V_o = 71 * V_m$ and the maximum voltage that the ADC-1 could measure is 29 volts ($V_m = 0.4095$). The resolution, or voltage increment for a change of one in the digital number, would be $(0.0001 * 71)$ or .071 volt. This circuit would draw 0.4 ma when measuring a power source of 29 volts.

Since most garden variety resistors are ± 5 or 10 percent, you should use precision resistors for any application where you desire greater than 5% accuracy.

6.0 Applications

The ADC-1 is a new product which has a diverse and varied future of applications. As Remote Measurement Systems receives reports from users on their installations, this chapter will grow. For the moment, this list will, hopefully, serve to illustrate some of the potential and actual applications for the ADC-1.

- * Build a weather station with data storage and graphical displays of temperature, wind speed, pressure, and rainfall.
- * Connect your security system to the ADC-1 for sophisticated alarm responses. Activate lighting upon sensing an intruder in your yard, while reserving a siren for an actual break-in.
- * Use the ADC-1 to control a star-tracker for your telescope or rotate solar panels to follow the sun (adds ~35% to energy collected).
- * Measure the electrical energy consumption of your home or appliances. If consumption exceeds a specific limit, your computer can alert you or even automatically shed part of the load. The price of the system could be paid for in a short time via lower electrical bills. Compare our price with a commercial watt-meter!
- * Operate a remote cabin from your primary residence. In seconds, via the telephone and a modem, you will know the status of the doors, windows, river level, and weather at your second home. Turn the furnace on before you plan on arriving!
- * Show your children how rapidly the freezer temperature rises on opening the door. Aid your child's introduction to computer uses by creating novel exploratory applications.
- * Monitor soil moisture of indoor plants and outdoor gardens. Use your computer to control garden watering or irrigation systems.
- * Sensors soon to be available will allow heart rate and blood pressure to be measured and plotted.
- * Experiment with robotics sensors and motor control.

- * Teach your students how computers are used in measuring and controlling processes in industry and let them invent their own uses in the school or home.
- * Collect data and control scientific experiments in the laboratory. The low power, small size, and ability to connect with a modem are ideal features for remote site data collection.
- * Control materials processing experiments while monitoring a variety of parameters.
- * Automate a steel fabricating plant by measuring and controlling the placement of rolling stock.

APPENDIX A

ALTERNATE PROTOCOLS FOR ADC-1/ COMPUTER INTERCONNECTION

This appendix contains instructions for connecting the ADC-1 to specific computers. Though short at this time, we will continue to add instructions as they become available. If the computer you are using is not listed, and you would like to contribute a short description of the required pin assignments for your system, we will include your experience in future editions of the manual.

VIC-20/COMMODORE 64

The ADC-1 has been designed to function with the VIC-20/Commodore 64 directly. The RS-232 compatible expansion module, needed for modems, is not required. The TTL levels used by the VIC-20 for its User Port are compatible with those employed by the ADC-1. Note that TTL level pins 21 and 22 on the ADC-1 DB-25 connector are used to connect the TRANSMITTED and RECEIVED DATA lines from the VIC-20. Since the full capacity of the VIC-20 regulated +5 volt power supply is available on pin 2 of the User Port, power interconnection with the ADC-1 is via pin 18 on the DB-25 connector. Be carefully not to short the power and ground wires from the VIC-20 as the VIC-20 power supply fuse can be easily blown. The Commodore Programmer's Reference Guide recommends that no more than 100mA be drawn from this pin. The User Port is discussed on pages 229 and 257 of the Guide.

The following pin connections should be made between the VIC-20 and the ADC-1 and will require four wires not to exceed 6 feet in length.

<u>ADC-1</u> <u>DB-25 pin #</u>	<u>SIGNAL</u>	<u>VIC-20/64</u> <u>User Port pin #</u>
22	Transmitted/Received Data	Both C and B
21	Received/Transmitted Data	M
7	Ground	N
18	Power (+5 v.)	2

In order for the VIC-20 port to function properly both pins B and C must be tied together and to ADC-1 pin 22.

ATARI 400/800

The Atari 400/800 computers are similar to the VIC-20 in their use of TTL levels on the serial I/O port. Connect the ADC-1 DB-25P to the ATARI serial bus connector (on the right-hand side of the Atari) as follows:

<u>ADC-1</u> <u>DB-25 pin #</u>	<u>SIGNAL</u>	<u>ATARI</u> <u>Serial Bus pin #</u>
22	Transmitted/Received Data	3
21	Received/Transmitted Data	5
7	Ground	4
18	Power (+5 v.)	10

S-100 Z80A System using the SSM IO-4 Interface Board

There are two headers J1 and J2 that serves as the connection points for the two RS-232 serial ports of the IO-4. Connect the ADC-1 as follows:

<u>SSM IO-4 (J1 or J2)</u>	<u>Signal</u>	<u>ADC-1 (DB-25)</u>
1	Received/Transmitted Data	2
11	Transmitted/Received Data	3
7	Ground	7
5	Power +12 V	9

Remember to set the parity off, and the number of stop bits to 2 using either SW1 or SW2 on the IO-4. The status port, when strapped on header W1 or W2, functions well with Output Data Available(ODA) tied to Data Bus D1. The computer should then mask for bit 2 when checking the status port for incoming data availability (see statements 20, 30 and 5200 in the demonstration programs of chapter 3). Set the baud rate on the ADC-1 to match that strapped on the IO-4 baud rate header.

NORTHSTAR HORIZON

The Northstar is equipped with two RS-232 serial ports. Port B may be used to connect to the ADC-1. The pin connections between the Northstar DB-25 and the ADC-1 DB-25 are as follows:

<u>NORTHSTAR PORT B</u>	<u>Signal</u>	<u>ADC-1 (DB-25)</u>
2	Received/Transmitted Data	2
3	Transmitted/Received Data	3
7	Ground	7
9	Power +12 V	18

Since the current available at +12v on pin 9 of the Northstar is limited to 20 ma., this supply can be connected to the +5v input of the ADC-1. A 5.6 volt zener in the ADC-1 is able to lower the voltage for the 14 ma. current draw of the ADC-1.

Port B of the Northstar is hardwired for operation as logical data port 4, logical status port 5, and an incoming data bit mask of 2.

OSBORNE I

The Osborne 1 comes equipped with one RS-232 port, although it does have two connectors, a 9 pin DB-9 and a DB-25 socket. Either may be used for communications with the ADC-1, however, the pin connections are different for each.

The 25 pin DB-25 connector may be wired as follows:

<u>ADC-1 DB-25 pin #</u>	<u>SIGNAL</u>	<u>OSBORNE 1 printer port pin #</u>
2	Transmitted/Transmitted Data	2
3	Received/Received Data	3
7	Ground	7
18	Power +5 V	5 and/or 6

There is no specific pin assigned for power on the Osborne 25 pin connector, but the low power requirements of the ADC-1 permit use of the small DC current available from the Clear to Send (CTS) and Data Set Ready (DSR) lines. Though the ADC-1 will function connected to either pin 5 or 6, the current available on each is limited internally, and it is recommended that both be used. To combine the two sources

use a small diode (such as a 1N4148 - \$.02 each) in series with both pin 5 and pin 6 before combining the two. This will prevent currents flowing back into one of the lines (CTS or DSR) from the other and possibly damaging the Osborne.

The 9 pin connector may be wired as follows:

<u>ADC-1</u> <u>DB-9 pin #</u>	<u>SIGNAL</u>	<u>OSBORNE 1</u> <u>modem port pin #</u>
2	Transmitted/Received Data	6
3	Received/Transmitted Data	2
7	Ground	1
9	Power +12 V	7

Since the Osborne 1 DB-9 connector has a +12 V supply pin this can be connected directly to pin 9 of the ADC-1 and should be able to supply up to 200 ma. Further explanations of the Osborne 1 RS-232 and modem ports may be found on pages 277-280 of the Osborne 1 Users Reference Guide.

OSBORNE EXECUTIVE

The Osborne Executive comes equipped with two separately addressable RS-232 ports, a modem and a printer port. Either may be used for communications with the ADC-1, however, the pin connections and the program port addresses are different for each.

The Printer port may be wired as follows:

<u>ADC-1</u> <u>DB-25 pin #</u>	<u>SIGNAL</u>	<u>OSBORNE EXECUTIVE</u> <u>printer port pin #</u>
2	Transmitted/Transmitted Data	2
3	Received/Received Data	3
7	Ground	7
18	Power +5 V	5
		and/or 6

There is no specific pin assigned for power on the Osborne Executive printer port connector, but the low power requirements of the ADC-1 permit use of the small DC current available from the Clear to Send (CTS) and Data Set Ready (DSR) lines. Though the ADC-1 will function connected to either pin 5 or 6, the current available on each is limited internally, and it is recommended that both be used. To combine the two sources use a small diode (such as a 1N4148 - \$.02 each) in series with both pin 5 and pin 6 before combining the two. This will prevent currents flowing back into one of the lines (CTS or DSR) from the other and possibly damaging the Osborne.

The Modem port may be wired as follows:

<u>ADC-1</u> <u>DB-25 pin #</u>	<u>SIGNAL</u>	<u>OSBORNE EXECUTIVE</u> <u>modem port pin #</u>
2	Transmitted/Received Data	3
3	Received/Transmitted Data	2
7	Ground	7
9	Power +12 V	11

Since the Executive modem port connector has a +12 V supply pin this can be connected directly to pin 9 of the ADC-1.

In addition, for operation of either port on the Osborne Executive, several pins must be wired together in order to simulate the handshaking which the Osborne expects and the ADC-1 does not need nor provide. Both the Request to Send (RTS) and the Data Terminal Ready (DTR) lines must be held high (+5V) to permit communications. The easiest way is to connect them to their counterparts which are always held high on the Osborne. Thus the following pin connections should be made on the Osborne DB-25:

4 (Request to Send - RTS) to 5 (Clear to Send - CTS)
20 (Data Terminal Ready - DTR) to 6 (Data Set Ready - DSR).

EPSON HX-20

The Epson HX-20 is equipped with an eight pin DIN socket labelled "RS-232" which can be used for interconnection with the ADC-1. You will need to obtain an eight pin male DIN connector to fit the socket and serve as the Epson end of the Epson/ADC-1 cable. These connectors are difficult to locate but can be ordered from Cal Switch, 13717 S. Normandie, Gardena, CA 213-770-2330 for about \$3.25 (Part # 15GM8M) and are manufactured by Switchcraft.

Connect the pins in the ADC-1 DB-25 to the Epson 8-pin DIN as follows:

<u>ADC-1</u> <u>DB-25 pin #</u>	<u>SIGNAL</u>	<u>EPSON HX-20</u> <u>RS-232 DIN pin #</u>
2	Transmitted/Received Data	3
3	Received/Transmitted Data	2
7	Ground	1
9	Power (+12 v.)	battery or AC adaptor

Epson advises not to obtain ADC-1 power directly from the HX-20 batteries. However, the AC adaptor for the Epson supplies 600 ma @ 6 VDC and can be tapped for ADC-1 power. For portable or remote applications, an independent battery supply (3 "D" cells will last > 190 hrs.) as discussed in Chapter 4 will function well.

IBM-Personal Computer

Many vendors have produced multi-purpose cards for the IBM-PC which include serial ports. The RS-232 pin assignments for interconnection with the ADC-1 have been reported for the following cards:

ComboPlus from AST Research, Irvine, Calif.

<u>ADC-1</u> <u>DB-25 pin #</u>	<u>SIGNAL</u>	<u>IBM DB-25</u> <u>pin #</u>
2	Transmitted/Received Data	2
3	Received/Transmitted Data	3
7	Ground	7
18	Power (+5 v.)	jumper from +5 volts on card to pin 13 (not otherwise used) or external supply

The Clear to Send (CTS) and Data Set Ready (DSR) lines are current limited to about 1 ma and do not provide sufficient power for ADC-1 operation. A jumper from +5 volt on the card may be connected to pin 13 of the card DB-25.

Asynchronous Communications Adaptor from IBM.

SCP-130 from Seattle Computer Products, Seattle, WA.

Persyst Spectrum Personal Systems Technology, Laguna Hills, CA.

<u>ADC-1</u> <u>DB-25 pin #</u>	<u>SIGNAL</u>	<u>IBM DB-25</u> <u>pin #</u>
2	Transmitted/Received Data	3
3	Received/Transmitted Data	2
7	Ground	7
18	Power (+5 v.)	jumper from +5 volts on card to pin 13 (not otherwise used) or external supply.

On the IBM and SCP cards, the Clear to Send (CTS) and Data Set Ready (DSR) lines are current limited to about 1 ma and do not provide sufficient power for ADC-1 operation. A jumper from +5 volts on the card may be connected to pin 13 of the card DB-25 in order to provide power for the ADC-1. The Persyst card has provision via a jumper described in the Persyst manual to supply +5 volts to pin 8 of the DB-25. This pin should then be connected to pin 18 of the ADC-1.

KAYPRO 2,4,10

The Kaypro RS-232 modem port may be wired as follows:

<u>ADC-1</u> <u>DB-25 pin #</u>	<u>SIGNAL</u>	<u>OSBORNE EXECUTIVE</u> <u>printer port pin #</u>
2	Transmitted/Transmitted Data	3
3	Received/Received Data	2
7	Ground	7
9	Power +12 V	4
		and 20

There is no specific pin assigned for power on the Kaypro modem port connector, but the low power requirements of the ADC-1 permit use of the small DC current available from the Request to Send (RTS) and Data Terminal Ready (DTR) lines. Though the ADC-1 will function connected to either pin 4 or 20, the current available on each is limited internally, and it is recommended that both be used. To combine the two sources use a small diode (such as a 1N4148 - \$.02 each) in series with both pin 4 and pin 20 before combining the two. This will prevent currents flowing back into one of the lines (RTS or DTR) from the other and possibly damaging the Kaypro.

APPENDIX B

BASIC PROGRAMS FOR SPECIFIC COMPUTERS

This appendix contains programming examples for operating the ADC-1 with specific computers. The programs are written in BASIC and have been successfully run on the respective computers. Each manufacturer has used a slightly different method and associated BASIC instruction to input and output single bytes to an external device such as the ADC-1.

To run the ADC-1 with your computer, refer to the listings in Chapter 3 and replace the subroutines beginning at statement 5000 with those listed here. Though on the block of instructions from 5000 to 6000 are listed here, the input/output instructions in the later subroutines should be altered to reflect the code appropriate for your computer.

Since several common computers do not support the logical AND and OR statements, it is necessary to use alternate coding to examine the state of a selected bit. One method is to use integer arithmetic. For example, if you want to examine bit 5 which has a decimal value of 16 (see the chart in Appendix C), the following code may be used:

```
X = BYTEIN           :let X equal the byte to be checked
S = INT(X/16)         :find the number of integer
                      :multiples of 16 that fit into X
IF INT(S/2)*2=S THEN BIT5=0 ELSE BIT5=1
                      :if S is even then bit 5 must be
                      :zero. Only if S is odd can Bit 5
                      :be one.
```

By replacing 16 in the second statement with the decimal value of the desired bit, any bit within a byte can be extracted and examined. This type of programming is required for Atari computers.

COMMODORE VIC-20

The program which reads the onboard temperature sensor as described in Section 3.4 using Microsoft Basic is altered for the VIC-20 as follows:

```
10 OPEN2,2,0,CHR$(136)+CHR$(0)      ;open I/O port#2 @ 1200 baud,
                                      ;no parity, 2 stop bits,
                                      ;3 wire communications.
20 GET#2,X$                          ;clear port of any old characters
30 CN=24                             ;set channel for ADC-1 temp sensor
40 GOSUB 5000                        ; goto channel start/read subroutine
61 REM                               ; on return will have millivolts*10
70 DK= Z/10                          ; calc temp in Kelvin = millivolts
80 DF=(9/5*(DK-273))+32              ; convert to degrees Fairheit
90 PRINT "TEMP= ",DF                 ; print temperature
100 GOTO 40                           ; repeat forever

5000 PRINT#2,CHR$(CN);                ; select temp sensor; start A/D conversion
5010 GOSUB 5200                        ; wait for ADC-1 response to start command
5020 PRINT#2,CHR$(161);                ; tell ADC-1 to return A/D high byte/status
5030 GOSUB 5200                        ; wait for reply
5040 HB=CH                             ; save high byte from A/D
5050 IF (HB AND 128)<>0 THEN 5020      ; check status for A/D finished
5060 PRINT#2,CHR$(145);                ; command ADC-1 to return A/D low byte
5070 GOSUB 5200                        ; wait for reply and read byte
5080 LB=CH                             ; read low byte from A/D into LBYTE
5085 HM=HB AND 15                      ; mask for 4 high order bits from A/D
5090 Z=LB+256*HM                       ; combine all 12 bits from A/D
5100 IF (HB AND 16)=0 THEN Z=-Z        ; fix sign if negative flag set
5110 RETURN

5200 GET#2,X$                          ; get character from buffer
5210 IF LEN(X$)=0 THEN GOTO 5200      ; if no character then try again
5215 CH=ASC(X$)                       ; convert string to ASC value of character
5220 RETURN                           ; return to calling program
```

Statement 10 is used to set the baud rate of the VIC I/O port and to define what logical number (#n) will be used to address the port from your program. The maximum baud rate possible with the VIC is 2400. This statement also sets 2 stop bits, 8 bit word length, no parity, full duplex, and 3 wire handshaking. A full explanation of this command is located in pages 252-254 of the VIC-20 Programmer's Reference Guide.

It is important to follow all PRINT#2,CHR\$(x) commands with a ";". This tells the VIC not to send a return following the character. The ADC-1 is expecting only one byte and will not function properly when it receives two closely spaced bytes.

The routine at statement 5200, which waits for the ADC-1 response, asks for a character from the input buffer, and checks for its length, which is zero for an empty buffer. Since the GET# command will only read characters into a string variable, the incoming byte must be converted to its decimal equivalent before the temperature can be calculated. Note that in VIC BASIC, only the first two letters of a variable name are significant and the variable names used here have been shortened from those used in Section 3.4. Also the comments in lower case should not be entered into your VIC-20.

COMMODORE 64

The Commodore 64 is very similar to the VIC-20 and since most of the comments in the VIC-20 section apply to the C64, you should read that section. However, the C64 has been found to occasionally miss part of the response byte sent by the ADC-1 after a command from the C64. In C64 serial communications, 10 bits are actually transferred for each byte of information sent. Eight are data bits followed by two stop bits. Infrequently, the C64 receives only 9 and appears to "hang-up" as it waits for the 10th bit.

The following program checks the RS232 Input Bit Count memory location(168) for ten bits. If, after a reasonable wait, it has not received all ten bits, the C64 re-transmits the original command and tries again. This program differs from the VIC-20 program in that the command to the ADC-1 is sent in the subroutine at 5200 rather than in the subroutine at 5000. Should you want to communicate with the ADC-1 at 300 baud, the timeout interval as set in statement 5220, should be changed from 3 to 12.

```

10 OPEN2,2,0,CHR$(136)+CHR$(0)      ;open I/O port#2 @ 1200 baud,
                                     ;no parity, 2 stop bits,
                                     ;3 wire communications.
20 GET#2,X$                          ;clear port of any old characters
30 CN=24                             ;set channel for ADC-1 temp sensor
40 GOSUB 5000                        ; goto channel start/read subroutine
61 REM                               ; on return will have millivolts*10
70 DK= Z/10                          ; calc temp in Kelvin = millivolts
80 DF=(9/5*(DK-273))+32              ; convert to degrees Fairheit
90 PRINT "TEMP= ",DF                 ; print temperature
100 GOTO 40                           ; repeat forever

```

```

5000 CMD=CHR$(CN);           ; select temp sensor; start A/D conversion
5010 GOSUB 5200               ; wait for ADC-1 response to start command
5020 CMD=CHR$(161);          ; tell ADC-1 to return A/D high byte/status
5030 GOSUB 5200               ; wait for reply
5040 HB=CH                    ; save high byte from A/D
5050 IF(HB AND 128)<>0 THEN 5020 ; check status for A/D finished
5060 CMD=CHR$(145);          ; command ADC-1 to return A/D low byte
5070 GOSUB 5200               ; wait for reply and read byte
5080 LB=CH                    ; read low byte from A/D into LBYTE
5085 HM=HB AND 15             ; mask for 4 high order bits from A/D
5090 Z=LB+256*HM              ; combine all 12 bits from A/D
5100 IF (HB AND 16)=0 THEN Z=-Z ; fix sign if negative flag set
5110 RETURN

5200 PRINT#2,CMD              ; get character from buffer
5205 TM=0                     ; set timeout to zero
5210 GET#2,X$                 ; get char, set ST flag if input char rec'd
5215 IF PEEK(168)<10 THEN TM=TM+1 ; increment timeout count as long as
                                ; input bit count is less than 10
5220 IF TM>3 THEN 5200         ; if loop is repeated 3 times, then C64
                                ; has failed to get input byte, so send
                                ; command again
5225 IF ST=0 THEN GOTO 5210    ; if no character then try again
5230 CH=PEEK(170)             ; get input character sent by ADC-1
5235 RETURN                    ; return to calling program

```

Additional comments regarding this program may be found following the VIC-20 program code.

KAYPRO 2,4,10

The Kaypro series of computers use I/O ports as described in Chapter 3 of this manual. The following lines of code which set the proper port numbers should be substituted for those listed in chapter 3.

```

10  PN=4                      :REM serial data port address
20  S=6                       :REM serial status port address
30  M=1                       :REM received byte mask

```

Baud rates on the Kaypro can be set by outputting a hex code to port zero (0). The correct statement is OUT 0,&HE for 9600 baud. Other baud rates are 300 - &H5, 600 - &H6, 1200 - &H7, 2400 - &HA and 4800 - &HC. Baud rates are discussed in the Kaypro User's Guide on page 67.

ATARI 400/800

Atari computers are capable of communication with the ADC-1 via the PEEK and POKE commands. It is first necessary to set the baud rate and serial mode controls by POKE'ing values to the requisite memory location. Configuring the Atari for operation with the ADC-1 at 1200 baud is done by the following statements:

```
10 POKE 53768,40      ;Clock channel 3 with 1.79 Mhz.
20 POKE 53764,234     ;Set baud rate to 1200 baud
30 POKE 53766,2
40 POKE 53770,0       ;Reset error flags
50 POKE 53775,35      ;Set serial mode control
```

It is now possible to operate the ADC-1 with your Atari. Commands should be POKE'd to memory location 53773 to send commands to the ADC-1. To check for incoming ADC-1 responses, location 53770 should be PEEK'd.

To send a command byte to the ATARI, do a

```
POKE 53773,command
```

To get data from the ADC-1 use

```
X=PEEK(53770)
```

To determine if data has been received from the serial input port use:

```
100 S=INT(PEEK(53774)/32)      ;read IRQST register
105 IF INT(S/2)*2 = S THEN 100 ;mask out serial input ready bit
110 X=PEEK(53770)              ;read character
```

This code uses integer arithmetic to check for individual bit values in place of the logical operator AND which is not implemented on the Atari. The loop at 100 and 105 is checking bit 6 (decimal 32) of the IRQST register. When this bit is 0, the register is empty. When a character has arrived, the bit goes to one and the character is read from the input memory location. This loop can be substituted for the loop at 5200 in the chapter 3 demonstration programs.

IBM-Personal Computer

The following code should be used with the IBM Personal Computer. The OPEN statement is used to configure the COM1 port for ADC-1 operation as logical I/O unit #1. Other than the OPEN, INPUT\$ and PRINT #1 statements, the code is identical to that discussed in chapter 3.

```
10 OPEN "COM1:9600,N,8,2,CS,DS" AS 1 ;open I/O port#1
                                ;@ 9600 baud, no parity, 8 bits,
                                ;2 stop bits, turn off handshaking
20 IF LOC(1)>0 THEN X$=INPUT$(LOC(1),1)
                                ;clear port of any old characters
30 CN=24                        ;set channel for ADC-1 temp sensor
40 GOSUB 5000                  ;goto channel start/read subroutine
61 REM                        ; on return will have millivolts*10
70 DK= Z/10                    ; calc temp in Kelvin = millivolts
80 DF=(9/5*(DK-273))+32        ; convert to degrees Fahrenheit
90 PRINT "TEMP= ",DF           ; print temperature (only first
                                ; decimal place is significant)

100 GOTO 40                    ; repeat forever

5000 PRINT #1,CHR$(CN);        ; select temp sensor; start A/D conversion
5010 GOSUB 5200                ; wait for ADC-1 response to start command
5020 PRINT #1,CHR$(161);       ; tell ADC-1 to return A/D high byte/status
5030 GOSUB 5200                ; wait for reply
5040 HB=CH                     ; save high byte from A/D
5050 IF (HB AND 128)<>0 THEN 5020 ; check status for A/D finished
5060 PRINT #1,CHR$(145);       ; command ADC-1 to return A/D low byte
5070 GOSUB 5200                ; wait for reply and read byte
5080 LB=CH                     ; read low byte from A/D into LBYTE
5085 HM=HB AND 15              ; mask for 4 high order bits from A/D
5090 Z=LB+256*HM               ; combine all 12 bits from A/D
5100 IF (HB AND 16)=0 THEN Z=-Z ; fix sign if negative flag set
5110 RETURN

5200 IF LOC(1)=0 THEN 5200      ; if no byte received, then try again
5210 X$=INPUT$(LOC(1),1)       ; get character from buffer
5215 CH=ASC(X$)                ; convert string to ASC value of character
5220 RETURN                    ; return to calling program
```

It should be possible to use the BASIC statements INP(pn) and OUT pn,x with the IBM. In this case the interrupt generated by the response byte from the ADC-1 would need to be serviced using a statement of the type ON COM GOTO xxx. However, Remote Measurement Systems has not yet been able to determine the port number. Since this alternate program would be much faster than the above illustration, any assistance from IBM programming specialists would be most welcome.

OSBORNE I

The method by which bytes are input and output by the Osborne 1 operating system is somewhat more complicated than those used by other computer systems discussed in this Appendix. The I/O functions are memory mapped in Bank 2 of memory while BASIC utilizes Bank 1. In order to send a byte to the ADC-1, Bank 2 must be swapped into use. After the command byte is transmitted and the ADC-1 response received, Bank 1 must be returned for proper operation of BASIC.

Remote Measurement Systems has written an assembly language subroutine to perform the above operations. This subroutine can be POKE'd into memory directly from BASIC and leaves a simple and clean program for the user. The program which reads the onboard temperature sensor as described in Section 3.4 using Microsoft Basic is altered for the Osborne 1 as follows:

```
10 ADC=VARPTR(#3):A=ADC      ;get address of I/O logical
                              ;file #3. This memory area used
                              ;to store RMS ADC-1 I/O routine.
13 FOR I=1 TO 32             ;loop to put routine into memory
14 READ X                    ;get byte of subroutine from DATA
                              ;statement at line number 9000
15 POKE A,X                  ;puts piece of subroutine in memory
                              ;starting at file #3 locations
16 A=A+1                     ;increment memory location
17 NEXT                       ;end loop after 32 bytes
20 R%=0                       ;clear integer to hold ADC-1 response
30 C%=24                      ;set channel for ADC-1 temp sensor
40 GOSUB 5000                 ; goto channel start/read subroutine
70 DK= Z/10                   ; calc temp in Kelvin = millivolts
80 DF=(9/5*(DK-273))+32       ; convert to degrees Fairenheit
90 PRINT "TEMP= ",DF          ; print temperature
100 GOTO 40                   ; repeat forever

5000 CALL ADC(C%,R%)          ; select temp sensor; start A/D conversion
                              ; C% contains command to ADC-1, R%
                              ; contains response. Output/input
                              ; routine ADC does the work.
5020 C%=161                   ; tell ADC-1 to return A/D high byte/status
5030 CALL ADC(C%,R%)          ; send command and get response
5040 HB=R%                     ; save high byte from A/D
5050 IF (HB AND 128)<>0 THEN 5020 ; check status for A/D finished
5060 C%=145                     ; command ADC-1 to return A/D low byte
5070 CALL ADC(C%,R%)          ; wait for reply and read byte
5080 LB=R%                     ; read low byte from A/D into LBYTE
5085 HM=HB AND 15              ; mask for 4 high order bits from A/D
5090 Z=LB+256*HM               ; combine all 12 bits from A/D
5100 IF (HB AND 16)=0 THEN Z=-Z ; fix sign if negative flag set
5110 RETURN

9000 DATA &HF3,&HD3,&H0,&H3E,&H0,&H32,&H8,&HEF,&H7E,&H32,&H1,
9005 DATA &H2A,&H3A,&H0,&H2A,&HE6,&H1,&H28,&HF9,&H3A,&H1,&H2A,
9010 DATA &H12,&HD3,&H1,&H3E,&H1,&H32,&H8,&HEF,&HFB,&HC9
```

The subroutine which does the input/output control is named "ADC" and is loaded from the DATA statements by the loop between statements 10 and 17. Once loaded, it does not need to be altered. When you include these statements in your own applications program be sure to impliment these instructions before trying to access the ADC-1. Also do not OPEN or CLOSE logical file#3 elsewhere in your program as it will erase the ADC-1 assembly language routine. Aside from this change, the program is quite similar to that discussed previously in the manual.

The baud rate of the port can be configured by using the CP/M utility "SETUP" to set the baud rate to that of the ADC-1. The protocol should be set to "XON/XOFF". No initialization string is required and the "OTHER DEVICE" response should be given to the device selection query.

The actual assembly language code follows for those users who are adept at assembly language programming and who may wish to alter our subroutine.

```

3:
4:                ;Subroutine to operate ADC-1 from Osborne I using Microsoft
5:                ;Basic
6:                ; CALL ADC(O%,I%) where
7:                                ; O% is the output char to ADC-1 and
8:                                ; I% is the input char from ADC-1.
9:
10: 2A01            sio      equ 2A01h
11: 2A00            sioc     equ 2A00h
12: EF08            flpos    equ 0EF08h
13:
14: 0000' F3        io: di                ;switch to bank 2
15: 0001' D3 00      out (0),a            ;interrupts are disabled
16: 0003' 3E 00      ld a,0
17: 0005' 32 EF08    ld (flpos),a
18: 0008' 7E        ld a,(hl)            ;get argument passed from basic
19: 0009' 32 2A01    ld (sio),a          ;send character to ADC-1
20: 000C' 3A 2A00    lop: ld a,(sioc)     ;wait for input character from ADC-1
21: 000F' E6 01      and 1
22: 0011' 2B F9      jr z,lop            ;char received ??
23: 0013' 3A 2A01    ld a,(sio)          ;yes - read input character
24: 0016' 12        ld (de),a            ;return in Basic variable
25: 0017' D3 01      out (1),a            ;reset memory banks
26: 0019' 3E 01      ld a,1
27: 001B' 32 EF08    ld (flpos),a
28: 001E' FB        ei
29: 001F' C9        ret
30:                end

```

OSBORNE EXECUTIVE

Programming for the the Executive also requires the use of an assembly language routine. Bytes can be transmitted using the simple BASIC command "OUT pn,cmd" as has been described in Chapter 3 of this manual. However, it is not possible to get the response byte value directly from BASIC due to the way the Executive services interrupts when a byte is received at the port. The Executive is an interrupt driven machine, and no method has yet been found to turn off the interrupts for a port.

We have written an assembly language routine which correctly receives data from the ADC-1 and installs within a user program very much like the subroutine "ADC" in the Osborne 1 example. Since the assembly language routine is somewhat long and complex, you should contact Remote Measurement Systems for a Osborne Executive diskette containing this subroutine and a more detailed explanation.

EPSON HX-20

The program which reads the onboard temperature sensor as described in Section 3.4 using Microsoft Basic is altered for the Epson HX-20 as follows:

```
10  OPEN"O",#1,"COM0:(68N2F)"      ;open I/O port#1 for output
                                   ;@ 4800 baud, 8 bit words,
                                   ;no parity, 2 stop bits,
                                   ;ignore handshaking
15  OPEN"I",#2,"COM0:(68N2F)"      ;open I/O port#2 for input
18  WIDTH "COM0:",255              ;set port line length to 255 so no
                                   ;carrage returns are sent
20  X$=INPUT$(LOF(2),#2)           ;clear port of any old characters
30  CN=24                          ;set channel for ADC-1 temp sensor
40  GOSUB 5000                     ; goto channel start/read subroutine
61  REM                            ; on return will have millivolts*10
70  DK= Z/10                       ; calc temp in Kelvin = millivolts
80  DF=(9/5*(DK-273))+32           ; convert to degrees Fairenheit
90  PRINT "TEMP= ",DF              ; print temperature
100 GOTO 40                        ; repeat forever

5000 PRINT#1,CHR$(CN);             ; select temp sensor; start A/D conversion
5010 GOSUB 5200                     ; wait for ADC-1 response to start command
5020 PRINT#1,CHR$(161);             ; tell ADC-1 to return A/D high byte/status
5030 GOSUB 5200                     ; wait for reply
5040 HB=CH                          ; save high byte from A/D
5050 IF (HB AND 128)<>0 THEN 5020    ; check status for A/D finished
5060 PRINT#1,CHR$(145);             ; command ADC-1 to return A/D low byte
5070 GOSUB 5200                     ; wait for reply and read byte
5080 LB=CH                          ; read low byte from A/D into LBYTE
5085 HM=HB AND 15                   ; mask for 4 high order bits from A/D
5090 Z=LB+256*HM                    ; combine all 12 bits from A/D
5100 IF (HB AND 16)=0 THEN Z=-Z     ; fix sign if negative flag set
5110 RETURN

5200 X$=INPUT$(LOF(2),#2)           ; get character from buffer
5210 IF LEN(X$)=0 THEN GOTO 5200    ; if no character then try again
5215 CH=ASC(X$)                     ; convert string to ASC value of character
5220 RETURN                          ; return to calling program
```

Statements 10 and 15 are used to set the mode and protocol of the input and output ports and to define what logical numbers (#n) will be used to direct incoming and outgoing bytes within your program. The input and output functions are refered to by separate port numbers on the Epson. The maximum baud rate possible with the Epson is 4800. This statement also sets 2 stop bits, 8 bit word length, no parity, and hankshaking. A full explanation of the OPEN command is located on page 306 of Volume 2 of the Epson Reference Manual.

A DATA ACQUISITION AND CONTROL SYSTEM WITH TERMINAL DISPLAY AND GRAPHING

```

1 REM This program demonstrates a fully functioning data acquisition and
2 REM control system for the ADC-1. Either a list or a graph may be displayed
3 REM on the terminal. An alarm system may be implemented when conditions and
4 REM responses are inserted after line 3000. The system was designed for an
5 REM S-100 Z-80A system using a Televideo 950 terminal. The program uses
6 REM cursor positioning to maintain a non-scrolling display.
7 REM
8 REM written by Keith Ronnholm, Remote Measurement Systems, Inc
9 REM P.O. Box 15544, Seattle, WA. 98115. 206-525-3369. 6/18/83
11 REM*****

15 DIM OD(71),LAB$(24),DS(6),TOFF(24),PON(24),NCHAN(24),NCHANCUR(24)
20 DIM ZAVG(17),ZSUM(17),DCNT(4),DLAB$(4)
30 ALARM=0 :REM initialize with alarm cond off
40 BYTEPT=0 :REM if 1 print hi & low A/D bytes
50 DIOPT=1 :REM digital inputs display option
51 REM 1 lists total counts, 2 displays on/off
60 FOR I=1 TO 6 : DS(I)=0 : NEXT :REM turn off all controlled outputs
70 FOR I=1 TO 70: OD(I)=44: NEXT :REM set initial graph line position
80 FOR I=1 TO 17
90 NCHAN(I)=5 :REM set # of times each channel is averaged
100 NCHANCUR(I)=0 :REM holds the current # within NCHAN
110 NEXT
200 REM -----
210 REM Temperature Offsets for sensors(TRAFF + TOFF)
220 TOFF(13)=-6
230 TOFF(12)=-2
240 TOFF(7)=-7
250 TOFF(1)=+4
260 TOFF(2)=+3
270 TOFF(24)=-10.1
280 BELL$=""
290 REM -----
300 LAB$(16)="COMP ROOM TEMP " :REM set up labels for display
310 LAB$(1)="FREEZER TEMP"
320 LAB$(2)="TEST TEMP #OR"
330 LAB$(3)="TEST TEMP #YELZ"
340 LAB$(4)="TEST TEMP #GRN"
350 LAB$(5)="TEST TEMP #BLU"
360 LAB$(6)="TEST TEMP #PUR"
370 LAB$(7)="LIVING ROOM TEMP"
380 LAB$(8)="ENERGY CONSUMP - W "
390 LAB$(9)="LIVING RM LIGHT"
400 LAB$(10)="ELECT USAGE - WATTS"
410 LAB$(11)="BASEMENT TEMP"
420 LAB$(12)="HOT TUB TEMP"
430 LAB$(13)="OUTSIDE TEMP"
440 LAB$(14)="WIND SPEED"
450 LAB$(15)="COMP ROOM LIGHT"
460 LAB$(17)="ONBOARD TEMP"

```

```

470 DLAB$(1)="Bedroom Vibrat"
480 DLAB$(2)=" Dig Input #2"
490 DLAB$(3)=" Refridge Door"
500 DLAB$(4)=" Dig Input #4"

610 INPUT "PLEASE ENTER WHICH PORT, A OR B, IS BEING USED: ",BDPT$
620 IF BDPT$="A" OR BDPT$="B" THEN GOTO 630 ELSE GOTO 610
630 IF BDPT$="A" THEN PN=4:S=5:M=2
640 IF BDPT$="B" THEN PN=6:S=7:M=2
645 X=INP(PN) :REM be sure port is clear of garbage
650 GOSUB 6000 :REM go reset digital inputs
660 GOSUB 8000 :REM go turn all controlled output off
670 PRINT CHR$(26), :REM erase screen before starting

800 PRINT:PRINT "THE FOLLOWING IS A LIST OF OPTIONS FOR USING THE RMS A/D BOARD"
810 FOR I=1 TO 17: PON(I)=1: NEXT :REM setup to display all channels
820 PRINT " 1 - LIST ALL CHANNELS IN SEQUENCE AND DISPLAY THE CHANNEL VALUE"
830 PRINT " 2 - SELECT A SPECIFIC CHANNEL (OR S) TO MONITOR"
840 PRINT " 3 - PLOT A SPECIFIC CHANNEL"
850 PRINT " 4 - SET ALARM SYSTEM ON/OFF"
860 PRINT " "
870 PRINT "THE ALARM CONDITIONS SET AT LINES 3000+ ARE NOW ",
880 IF ALARM=1 THEN PRINT "ON" ELSE PRINT "OFF"
890 PRINT
900 INPUT "PLEASE SELECT YOUR PREFERED OPTION NOW - ",ACT
910 ON ACT GOTO 1070,1010,4000,920
920 GOSUB 940: GOTO 670
930 REM *****

940 PRINT USING "&";CHR$(27),CHR$(61),CHR$(52),CHR$(32),
950 PRINT "TO SET OR TURN OFF THE CONDITIONS, "
960 INPUT "TYPE ON OR OFF; TO LEAVE THEM THE SAME, TYPE A RETURN. ",ALAR$
970 IF ALAR$="ON" OR ALAR$="on" THEN ALARM=1
980 IF ALAR$="OFF" OR ALAR$="off" THEN ALARM=0
990 RETURN
1000 REM *****

1010 FOR I=1 TO 17: PON(I)=0: NEXT :REM turn display of all channels off
1020 INPUT "WHICH CHANNEL DO YOU WISH TO MONITOR? ",CHAN
1030 PON(CHAN)=1 :REM select which channels to display
1040 INPUT "DO YOU WISH TO ADD ANOTHER CHANNEL ? ",ACH$
1050 IF ACH$="Y" THEN GOTO 1020
1060 REM -----

1070 PRINT CHR$(26) :REM erase screen
1080 PRNTL=0 :REM first time thru print labels
1090 CNT=CNT+1 :REM increment sample # counter
1100 PRINT " ANALOG INPUTS COUNT ",CNT,
1110 IF ALARM=1 THEN PRINT USING "&";"ALARM ON "
1120 IF ALARM=0 THEN PRINT USING "&";"ALARM OFF"
1130 FOR J=1 TO 17 :REM loop for 16 analog inputs + onboard temp
1140 IF PON(J)=1 THEN JA=J ELSE GOTO 1410:REM if in display list then do

```

```

1150 GOSUB 2010                                :REM go get value of channel
1160 IF PRNTL=1 THEN PRINT USING "&";CHR$(27),CHR$(61),CHR$(34+J),CHR$(62),:GOTO 1200
1170 IF PRNTL=0 THEN PRINT USING "&";CHR$(27),CHR$(61),CHR$(34+J),CHR$(32),
1180 PRINT USING "##";J,: PRINT USING "&";" ",:REM print channel number
1190 PRINT USING "Ⓢ";LAB$(J),:REM print channel label
1200 IF BYTEPT=0 THEN GOTO 1240                :REM if bytept=1 print A/D bytes
1210 PRINT USING "####";HMASK,
1220 PRINT USING "####";LBYTE,
1230 PRINT USING "&";" "
1240 PRINT USING "#####.##";ZT,:PRINT USING "&";" ",:REM print datum
1250 IF J=17 THEN PRNTL=1                    :REM after first listing don't print label

1260 IF NCHAN(J)=1 THEN GOTO 1340:REM if not time averaging then next channel
1270 ZSUM(J)=ZSUM(J)+ZT                      :REM add new value to current sum
1280 NCHANCUR(J)=NCHANCUR(J)+1               :REM increment # of current values in sum
1290 IF NCHANCUR(J)<NCHAN(J) THEN GOTO 1330   :REM print last avg value
1300 IF NCHANCUR(J)>NCHAN(J) THEN PRINT "ERROR": STOP:REM if cur # = avg #
1310 ZAVG(J)=ZSUM(J)/NCHAN(J)               :REM calculate new average value
1320 ZSUM(J)=0:NCHANCUR(J)=0                :REM reset counter and sum variables
1330 PRINT USING "#####.##";ZAVG(J),:REM print value averaged nchan samples

1340 PRINT USING "&";" "                      :REM at end of line print return
1350 IF J<16 THEN GOSUB 2510                 :REM print input status after each channel
1355 IF J<16 AND ALARM=1 THEN GOSUB 3000:REM if alarm flag on,check conditions

1360 NEWACT= INP(124)                       :REM get character from keyboard
1370 IF NEWACT=77 OR NEWACT=109 THEN GOTO 670 :REM if M, new menu
1380 IF NEWACT=71 OR NEWACT=103 THEN GOTO 4000 :REM if G, graph
1390 IF NEWACT=67 OR NEWACT=99 THEN GOSUB 1710 :REM if C, controlled out
1400 IF NEWACT=65 OR NEWACT=97 THEN GOSUB 940 :REM if A; change alarm stat

1410 NEXT J                                :REM goto next channel
1500 REM #####

1510 IF ACT=2 THEN PRINT " ":GOTO 1600 :REM if short list skip bottom labels
1520 PRINT USING "&";CHR$(27),CHR$(61),CHR$(54),CHR$(32),:REM position cursor
1530 PRINT USING "&";"Controlled Output Status:",
1540 FOR I=1 TO 6                          :REM print all 6 controlled outputs
1550 IF DS(I)=1 THEN PRINT USING "##";I,:PRINT USING "&";" ON "
1560 IF (DS(I)=0) THEN PRINT USING "##";I,:PRINT USING "&";" OFF "
1570 NEXT I: PRINT " "                     :REM return/line feed after last output

1580 PRINT USING "&";"M - NEW MENU; C - CONTROL CHANGE; G - GRAPH CHANNEL",
1590 PRINT USING "&";" "; A - ALARM CHANGE",
1600 PRINT USING "&";CHR$(27),CHR$(61),CHR$(32),CHR$(32),:REM top left screen
1610 IF ACT=2 THEN PRINT                   :REM if short list need extra return
1620 GOTO 1090                             :REM goto start of display cycle
1700 REM *****

1710 PRINT USING "&";CHR$(27),CHR$(61),CHR$(51),CHR$(32),
1720 PRINT:INPUT "WHICH CONTROLLED OUTPUT DO YOU WISH TO CHANGE? " ,K
1730 INPUT "ENTER A 1 FOR ON AND A 0 FOR OFF " ,DS(K)
1740 GOSUB 8000                             :REM send new controlled out configuration
1750 RETURN

```

```

2010 CN=JA-1 :REM set channel number for gosub
2020 IF CN=16 THEN CN=24 :REM set cn for onboard temp sensor
2030 X=INP(PN) :REM be sure port is clear of garbage
2040 GOSUB 5000 :REM go get digital number from port
2050 ZT=Z :REM Z in mv is printed if no calcs
2060 IF JA=10 OR JA=8 THEN GOSUB 2150 :REM if energy consumption, then watts
2070 IF JA=14 THEN LET ZT=Z*.03375 :REM wind speed calc - miles/hour
2080 IF JA>=1 AND JA<8 THEN GOSUB 2130 :REM go sub for temp conversions
2090 IF JA>=11 AND JA<=13 THEN GOSUB 2130
2100 IF CN=24 THEN GOSUB 2130
2110 RETURN

2129 REM converts degrees kelvin to fahrenheit and adds error correction
2130 LET ZT=(ABS(Z)/10-273.16)*1.8+32+TOFF(CN)
2140 RETURN
2150 ZUA=Z/1000: RAC=250: VAC=110 :REM mv to watts subroutine
2160 IF ZUA<7.5 THEN POR=ZUA :REM interpolation from graph
2170 IF ZUA>=7.5 THEN POR=(ZUA-7.5)*.00488 +.1 :REM in appendix D -A603
2180 IF ZUA>=28 THEN POR=(ZUA-28)*.00364 +.2 :REM call RMS for update on
2190 IF ZUA>=55.5 THEN POR=(ZUA-55.5)*.00348 +.3 :REM this section.
2200 IF ZUA>=199 THEN POR=(ZUA-199)*.00325 +.8
2210 ZT=POR*RAC*VAC :REM watts sensed
2220 RETURN

2500 REM *****
2510 PRINT USING "&";CHR$(27),CHR$(61),CHR$(52),CHR$(32), :REM go bottom page
2520 PRINT USING "&";"Digital Inputs: ",
2530 PRINT USING "Ⓢ";DLAB$(1),DLAB$(2),DLAB$(3),DLAB$(4),
2540 PRINT USING "&";" ",
2550 FOR I=1 TO 4 :REM check each input
2560 DTEST=2↑(I-1) :REM setup bit mask for each input
2570 ON DIOPT GOTO 2580,2610 :REM 1 - total counts; 2 - on/off
2580 IF DIGIN AND DTEST THEN DCNT(I)=DCNT(I)+1: REM IF ON THEN INCREMENT
2590 PRINT USING "#####";DCNT(I),
2600 GOTO 2630
2610 PRINT USING "&";" ", :REM display spacing
2620 IF DIGIN AND DTEST THEN PRINT USING "&";" ON", ELSE PRINT USING "&";"OFF",
2630 NEXT
2640 PRINT USING "&";" " :REM return/line feed at end
2650 RETURN

3000 REM ++++++
3002 REM In this section various conditions can be set and the reaction
3004 REM to them programmed. This example turns on multiple lights when
3006 REM Digital Input #1 goes ON. DI#1 is then reset.
3100 IF DIGIN AND 1 THEN GOSUB 3200 ELSE RETURN :REM if off, OK - no action
3110 GOSUB 6000 :REM reset latched inputs
3150 RETURN
3200 UC=204:GOSUB 7000 :REM address for BSR unit 1
3210 UC=197:GOSUB 7000 :REM turn it on
3220 UC=212:GOSUB 7000 :REM address for unit 4
3230 UC=197:GOSUB 7000 :REM turn it on
3240 UC=194:GOSUB 7000 :REM address for unit 5
3250 UC=197:GOSUB 7000 :REM turn it on
3260 RETURN

```

```

3998 REM This section is for a Televideo 950 Terminal Plot of any Channel
3999 REM It was written months ago without comments. Sorry. Ask for update.
4000 INPUT "WHICH CHANNEL DO YOU WISH TO PLOT? - ",JA
4010 ACT=3
4020 PMIN=0: PMAX=4095
4030 INPUT "HOW MANY SAMPLES DO YOU WISH TO AVERAGE PER PLOTTED POINT? ",NUMS
4040 DPT=1
4050 COL=DPT+10
4060 PRINT USING "&";"CURRENT PLOTTING LIMITS ARE ",:PRINT USING "####.#";PMIN,
4070 PRINT USING "&";" TO ",:PRINT USING "####.#";PMAX
4080 INPUT "DO YOU WISH TO CHANGE THESE? ",ANS$
4090 IF ANS$="Y" THEN GOSUB 4100 ELSE 4130
4100 INPUT "PLEASE ENTER MINIMUM LIMIT FOR PLOTTING - ",PMIN
4110 INPUT "PLEASE ENTER MAXIMUM LIMIT FOR PLOTTING - ",PMAX
4120 ICASC=COL+31
4130 INC=(PMAX-PMIN)/18
4140 S9$=SPACE$(9)
4150 SECPP=NUMS*.194561
4160 IF SECPP>45 THEN TYPEPLT=2 ELSE TYPEPLT=1
4170 MINPP=SECPP/60!
4180 PRINT USING "@          @";LAB$(JA),
4190 PRINT USING "&";"      #SAM/PT. ",:PRINT USING "####";NUMS,
4200 IF SECPP<60 THEN PRINT USING "&";"      SECS/PT ",:PRINT USING "####.#";SECPP ,
4210 IF SECPP>=60 THEN PRINT USING "&";"      MIN/PT ",:PRINT USING "####.#";MINPP,
4220 PRINT USING "&";"      COUNT ",:PRINT USING "####";CNT
4230 PRINT SPACE$(80)
4240 GOSUB 4260
4250 GOTO 4270
4260 PRINT USING "&";S9$, : FOR L=1 TO 70: PRINT USING "&";".",:NEXT:PRINT:RETURN
4270 FOR L=1 TO 9: AXIS=PMIN+(10-L)*INC*2!
4280 SP$=SPACE$(68)
4290 P$="."
4300 S5$=SPACE$(2)
4310 PRINT USING "####.#";AXIS,:PRINT USING "&";S5$,P$,SP$,P$
4320 PRINT USING "&";S9$,P$,SP$,P$
4330 NEXT
4340 GOSUB 4260
4350 PRINT SPACE$(80)
4360 PRINT USING "&";"C - CHANNEL;   S - SCALE;   N - #SAMPLES;   M - MENU;   L - LIST "
4370 COL=DPT+1
4380 SUM=0
4390 FOR I=1 TO NUMS
4400 GOSUB 2000
4410 SUM=SUM+ZT
4420 NEXT
4430 Z=SUM/NUMS
4440 CNT=CNT+1
4450 PRINT USING "&";CHR$(27),CHR$(45),CHR$(48),CHR$(32),CHR$(100),CHR$(32),
4460 PRINT USING "####";CNT
4470 IF Z<PMIN THEN Z=PMIN
4480 IF Z>PMAX THEN PMAX=Z*1.1: DPT=1: GOTO 4120
4490 ROW =(Z-PMIN)/INC
4500 IRASC=(23-ROW)+30
4510 ON TYPEPLT GOTO 4520,4630
4520 ORASC=OD(DPT)

```

```

4530 OD(DPT)=IRASC
4540 ICASC=COL+31+9
4550 CROW=ORASC
4560 CCOL=ICASC
4570 IF CROW=53 THEN CHAR=46 ELSE CHAR=32
4580 GOSUB 4750
4590 CROW=IRASC
4600 CHAR=42
4610 GOSUB 4750
4620 GOTO 4770
4630 OD(70)=IRASC
4640 FOR IN=2 TO 69
4650 CCOL=IN+31+9
4660 CROW=OD(IN)
4670 CHAR=32
4680 GOSUB 4750
4690 OD(IN)=OD(IN+1)
4700 CROW=OD(IN)
4710 CHAR=42
4720 GOSUB 4750
4730 NEXT
4740 GOTO 4770
4750 PRINT USING "&";CHR$(27),CHR$(45),CHR$(48),CHR$(CROW),CHR$(CCOL),CHR$(CHAR)
4760 RETURN
4770 DPT=DPT+1
4780 IF DPT>69 THEN DPT=DPT-69
4790 NEWACT=INP(124)
4800 IF NEWACT=83 THEN GOTO 4100
4810 IF NEWACT=67 THEN GOTO 4000
4820 IF NEWACT=78 THEN GOTO 4030
4830 IF NEWACT=77 THEN GOTO 670
4840 IF NEWACT=76 THEN ACT=1:GOTO 1070
4850 GOTO 4370
4860 WEND
4870 REM *****

```

```

5000 OUT PN,CN           :REM select temp sensor; start A/D conversion
5010 GOSUB 5200           :REM wait for ADC-1 response to start command
5020 DIGIN=CHIN          :REM store digital inputs status from response
5030 OUT PN,129+32       :REM tell ADC-1 to return A/D high byte/status
5040 GOSUB 5200           :REM wait for reply
5050 HBYTE=CHIN          :REM save high byte from A/D
5060 IF(HBYTE AND 128)<>0 THEN 5030 :REM check status for A/D finished
5070 OUT PN,129+16       :REM command ADC-1 to return A/D low byte
5080 GOSUB 5200           :REM wait for reply and read byte
5090 LBYTE=CHIN          :REM read low byte from A/D into LBYTE
5100 HMASK=HBYTE AND 15  :REM mask for 4 high order bits from A/D
5110 Z=LBYTE+256*HMASK    :REM combine all 12 bits from A/D
5120 IF (HBYTE AND 16)=0 THEN Z=-Z :REM fix sign if negative flag set
5130 RETURN

5200 IF(INP(S) AND M)=0 THEN 5200 :REM waits for response byte to come
5210 CHIN=INP(PN)         :REM get response from ADC-1
5220 RETURN               :REM return to calling program
5999 REM *****

6000 OUT PN,129          :REM read digital inputs without resetting
6010 GOSUB 5200           :REM wait for communications to be done
6020 D = CHIN AND 15      :REM mask for wanted input bits
6030 OUT PN,128          :REM reset latched inputs
6040 GOSUB 5200           :REM wait, read byte to clear port
6050 RETURN               :REM
6060 REM *****

7000 OUT PN,UC           :REM send unit code to ADC-1
7010 GOSUB 5200           :REM get ADC-1 response
7020 OUT PN,129          :REM ask for BSR busy/done status
7030 GOSUB 5200           :REM wait for answer
7040 IF (CHIN AND 16)<>0 THEN 7020 :REM if still busy, ask again
7050 RETURN               :REM done, return to main program
7060 REM *****

8000 DOUT=DS(1)+2*DS(2)+4*DS(3)+8*DS(4)+16*DS(5)+32*DS(6)+64
8010 OUT PN,DOUT         :REM send output control byte set in DOUT
8020 GOSUB 5200           :REM wait for communications to be done
8030 RETURN               :REM return to calling program

```

This program is currently available from Remote Measurement Systems on an 8" single sided, single density floppy disk for a nominal fee (the cost of a disk). Additional software and alternate disk formats will be available as they are developed. If you would like to obtain this program via telephone transfer through a 300 baud modem, call us to make arrangements.

C-1 - COMMAND AND RESPONSE BYTES - BIT ASSIGNMENTS

	MSB							LSB
Bit Number	8	7	6	5	4	3	2	1
Decimal Bit Value	128	64	32	16	8	4	2	1

The two most significant bits (8,7) of a command byte determine which of four possible actions the ADC-1 will take. Bits 6 and 5 select the contents of the response byte, even when those bits are used to specify a command activity. The response byte in some cases will not contain useful information and is treated as a reply indicating acceptance of the previous command by the ADC-1.

Bits 8,7 Select the following ADC-1 operations:

0,0 - Start the analog to digital converter on the specified channel.

Bit 6 is not used.

Bits 5,4 = 1,1 selects the onboard temperature sensor.

Bits 5,4 = 1,0 selects high gain instrumentation amplifier for analog channels 1-8 (specified by bits 1-3).

Bit 5 = 0, Bits 4 thru 1 encode analog input channels 1 to 16.
(decimal value of bits 1-4 ranges from 0 to 15).

0,1 - Sets state of the controlled outputs as specified by bits 1-6.

Outputs are numbered on the ADC-1 terminal block sequentially with bit 1 assigned to control output #1, bit 2 to output #2, etc.

A bit value of 1 encodes an ON, a bit value of 0 encodes OFF.

1,0 - When bit 1 = 0, this command causes the reset of the digital inputs configured for latched operation. Otherwise this setting of bits 8,7 does not initiate any ADC-1 operation (ie. NOP). Selection of the NOP is used to return digital input and status information.

1,1 - Send a command on the AC powerline to control an external device.

Bits 1 to 6 specify the BSR unit address and command codes.

Appendix C lists the 32 address and 6 control codes.

Bits 6,5 of the command byte cause the ADC-1 response byte to contain:

0,0 - Status of the four digital inputs.

Response byte bits 1 to 4 encode for the corresponding digital inputs #1 to #4. Bit 5 provides the status of the BSR transmitter (bit 5 = 0, ready; bit 5 = 1, unit busy transmitting).

0,1 - Least significant 8 bits from 12-bit A/D conversion (bit 1 is LSB).

1,0 - High byte from A/D conversion plus status information.

Response bits:

Bit 8 - A/D conversion status (0 = done; 1 = convert in progress).

Bit 7 - not used

Bit 6 - A/D voltage overrange (0 = valid reading; 1 = overvoltage).

Bit 5 - A/D polarity sign (0 = negative; 1 = positive input).

Bits 1-4 - most significant bits from the A/D (bit 4 is MSB).

1,1 - Garbage. response bits are tri-stated and meaningless.

C-2 - ADC-1 COMMAND INSTRUCTION SET

COMMAND Binary	BYTE VALUE Decimal	ADC-1 ACTION (initiated)	ADC-1 RESPONSE BYTE (contents)
00000000	0	Start A/D on input channel 1	Digital inputs/status
00000001	1	Start A/D on input channel 2	Digital inputs/status
00000010	2	Start A/D on input channel 3	Digital inputs/status
00000011	3	Start A/D on input channel 4	Digital inputs/status
00000100	4	Start A/D on input channel 5	Digital inputs/status
00000101	5	Start A/D on input channel 6	Digital inputs/status
00000110	6	Start A/D on input channel 7	Digital inputs/status
00000111	7	Start A/D on input channel 8	Digital inputs/status
00001000	8	Start A/D on input channel 9	Digital inputs/status
00001001	9	Start A/D on input channel 10	Digital inputs/status
00001010	10	Start A/D on input channel 11	Digital inputs/status
00001011	11	Start A/D on input channel 12	Digital inputs/status
00001100	12	Start A/D on input channel 13	Digital inputs/status
00001101	13	Start A/D on input channel 14	Digital inputs/status
00001110	14	Start A/D on input channel 15	Digital inputs/status
00001111	15	Start A/D on input channel 16	Digital inputs/status
00010000	16	Start A/D w/Inst Amp on ch# 1	no useful information
00010001	17	Start A/D w/Inst Amp on ch# 2	no useful information
00010010	18	Start A/D w/Inst Amp on ch# 3	no useful information
00010011	19	Start A/D w/Inst Amp on ch# 4	no useful information
00010100	20	Start A/D w/Inst Amp on ch# 5	no useful information
00010101	21	Start A/D w/Inst Amp on ch# 6	no useful information
00010110	22	Start A/D w/Inst Amp on ch# 7	no useful information
00010111	23	Start A/D w/Inst Amp on ch# 8	no useful information
00011000	24	Start A/D read onboard temp	no useful information
00011001	25	Start A/D read onboard temp	no useful information
00011010	26	Start A/D read onboard temp	no useful information
00011011	27	Start A/D read onboard temp	no useful information
00011100	28	Start A/D read onboard temp	no useful information
00011101	29	Start A/D read onboard temp	no useful information
00011110	30	Start A/D read onboard temp	no useful information
00011111	31	Start A/D read onboard temp	no useful information
00100000	32	Start A/D on input channel 1	no useful information
00100001	33	Start A/D on input channel 2	no useful information
00100010	34	Start A/D on input channel 3	no useful information
00100011	35	Start A/D on input channel 4	no useful information
00100100	36	Start A/D on input channel 5	no useful information
00100101	37	Start A/D on input channel 6	no useful information
00100110	38	Start A/D on input channel 7	no useful information
00100111	39	Start A/D on input channel 8	no useful information

<u>COMMAND</u> <u>Binary</u>	<u>BYTE VALUE</u> <u>Decimal</u>	<u>ADC-1 ACTION</u> <u>(initiated)</u>	<u>ADC-1 RESPONSE BYTE</u> <u>(contents)</u>
00101000	40	Start A/D on input channel 9	no useful information
00101001	41	Start A/D on input channel 10	no useful information
00101010	42	Start A/D on input channel 11	no useful information
00101011	43	Start A/D on input channel 12	no useful information
00101100	44	Start A/D on input channel 13	no useful information
00101101	45	Start A/D on input channel 14	no useful information
00101110	46	Start A/D on input channel 15	no useful information
00101111	47	Start A/D on input channel 16	no useful information
00110000	48	Start A/D w/Inst Amp on ch# 1	no useful information
00110001	49	Start A/D w/Inst Amp on ch# 2	no useful information
00110010	50	Start A/D w/Inst Amp on ch# 3	no useful information
00110011	51	Start A/D w/Inst Amp on ch# 4	no useful information
00110100	52	Start A/D w/Inst Amp on ch# 5	no useful information
00110101	53	Start A/D w/Inst Amp on ch# 6	no useful information
00110110	54	Start A/D w/Inst Amp on ch# 7	no useful information
00110111	55	Start A/D w/Inst Amp on ch# 8	no useful information
00111000	56	Start A/D read onboard temp	no useful information
00111001	57	Start A/D read onboard temp	no useful information
00111010	58	Start A/D read onboard temp	no useful information
00111011	59	Start A/D read onboard temp	no useful information
00111100	60	Start A/D read onboard temp	no useful information
00111101	61	Start A/D read onboard temp	no useful information
00111110	62	Start A/D read onboard temp	no useful information
00111111	63	Start A/D read onboard temp	no useful information
01000000	64	Set Control Outputs - All Off	Digital inputs/status
01000001	65	Set CO. - #1 on; #2,3,4,5,6 off	Digital inputs/status
01000010	66	Set CO. - #2 on; #1,3,4,5,6 off	Digital inputs/status
01000011	67	Set CO. - #1,2 on; #3,4,5,6 off	Digital inputs/status
01000100	68	Set CO. - #3 on; #1,2,4,5,6 off	Digital inputs/status
/	/	/	/
01001111	79	Set CO. - #1,2,3,4 on; #5,6 off	Digital inputs/status
01010000	80	Set CO. - #5 on; #1,2,3,4,6 off	A/D low order byte
01010001	81	Set CO. - #1,5 on; #2,3,4,6 off	A/D low order byte
/	/	/	/
01011111	95	Set CO. - #1,2,3,4,5 on; #6 off	A/D low order byte
01100000	96	Set CO. - #6 on; #1,2,3,4,5 off	A/D high byte & status
01100001	97	Set CO. - #1,6 on; #2,3,4,5 off	A/D high byte & status
/	/	/	/
01101111	111	Set CO. - #1,2,3,4,6 on; #5 off	A/D high byte & status

COMMAND Binary	BYTE VALUE Decimal	ADC-1 ACTION (initiated)	ADC-1 RESPONSE BYTE (contents)
01110000	112	Set CO. - #6 on; #1,2,3,4,5 off	meaningless garbage
01110001	113	Set CO. - #1,6 on; #2,3,4,5 off	meaningless garbage
/	/	/	/
01111110	126	Set CO. - #2,3,4,5,6 on; #1 off	meaningless garbage
01111111	127	Set Controlled Outputs - All ON	meaningless garbage
10000000	128	Reset latched digital inputs	Digital inputs/status
10000001	129	No ADC-1 operation	Digital inputs/status / BSR status *
10000010	130	Reset latched digital inputs	Digital inputs/status
10000011	131	No ADC-1 operation	Digital inputs/status
/	/	/	/
10001110	142	Reset latched digital inputs	Digital inputs/status
10001111	143	No ADC-1 operation	Digital inputs/status
10010000	144	Reset latched digital inputs	A/D low order byte
10010001	145	No ADC-1 operation	A/D low order byte
10010010	146	Reset latched digital inputs	A/D low order byte
10010011	147	No ADC-1 operation	A/D low order byte
/	/	/	/
10011110	158	Reset latched digital inputs	A/D low order byte
10011111	159	No ADC-1 operation	A/D low order byte
10100000	160	Reset latched digital inputs	A/D high byte & status
10100001	161	No ADC-1 operation	A/D high byte & status
10100010	162	Reset latched digital inputs	A/D high byte & status
10100011	163	No ADC-1 operation	A/D high byte & status
/	/	/	/
10101110	174	Reset latched digital inputs	A/D high byte & status
10101111	175	No ADC-1 operation	A/D high byte & status
10110000	176	Reset latched digital inputs	meaningless garbage
10110001	177	No ADC-1 operation	meaningless garbage
10110010	178	Reset latched digital inputs	meaningless garbage
10110011	179	No ADC-1 operation	meaningless garbage
/	/	/	/
10111110	190	Reset latched digital inputs	meaningless garbage
10111111	191	No ADC-1 operation	meaningless garbage
11000000	192	BSR address - unit 13, house 1	Digital inputs/status
11000001	193	BSR command - CLEAR	Digital inputs/status
11000010	194	BSR address - unit 5, house 1	Digital inputs/status
11000011	195	BSR command - ALL	Digital inputs/status
11000100	196	BSR address - unit 3, house 1	Digital inputs/status
11000101	197	BSR command - ON, house 1	Digital inputs/status
11000110	198	BSR address - unit 11, house 1	Digital inputs/status
11000111	199	BSR command - OFF, house 1	Digital inputs/status

COMMAND Binary	BYTE VALUE Decimal	ADC-1 ACTION (initiated)	ADC-1 RESPONSE BYTE (contents)
11001000	200	BSR address - unit 15, house 1	Digital inputs/status
11001001	201	BSR command - BRIGHT , house 1	Digital inputs/status
11001010	202	BSR address - unit 7, house 1	Digital inputs/status
11001011	203	BSR command - DIM , house 1	Digital inputs/status
11001100	204	BSR address - unit 1, house 1	Digital inputs/status
11001101	205	BSR address - unit 1, house 1	Digital inputs/status
11001110	206	BSR address - unit 9, house 1	Digital inputs/status
11001111	207	BSR address - unit 9, house 1	Digital inputs/status
11010000	208	BSR address - unit 14, house 1	A/D low order byte
11010001	209	BSR address - unit 14, house 1	A/D low order byte
11010010	210	BSR address - unit 6, house 1	A/D low order byte
11010011	211	BSR address - unit 6, house 1	A/D low order byte
11010100	212	BSR address - unit 4, house 1	A/D low order byte
11010101	213	BSR address - unit 4, house 1	A/D low order byte
11010110	214	BSR address - unit 12, house 1	A/D low order byte
11010111	215	BSR address - unit 12, house 1	A/D low order byte
11011000	216	BSR address - unit 16, house 1	A/D low order byte
11011001	217	BSR address - unit 16, house 1	A/D low order byte
11011010	218	BSR address - unit 8, house 1	A/D low order byte
11011011	219	BSR address - unit 8, house 1	A/D low order byte
11011100	220	BSR address - unit 2, house 1	A/D low order byte
11011101	221	BSR address - unit 2, house 1	A/D low order byte
11011110	222	BSR address - unit 10, house 1	A/D low order byte
11011111	223	BSR address - unit 10, house 1	A/D low order byte
11100000	224	BSR address - unit 13, house 2	A/D high byte & status
11100001	225	BSR command - CLEAR	A/D high byte & status
11100010	226	BSR address - unit 5, house 2	A/D high byte & status
11100011	227	BSR command - ALL	A/D high byte & status
11100100	228	BSR address - unit 3, house 2	A/D high byte & status
11100101	229	BSR command - ON, house 2	A/D high byte & status
11100110	230	BSR address - unit 11, house 2	A/D high byte & status
11100111	231	BSR command - OFF, house 2	A/D high byte & status
11101000	232	BSR address - unit 15, house 2	A/D high byte & status
11101001	233	BSR command - BRIGHT , house 2	A/D high byte & status
11101010	234	BSR address - unit 7, house 2	A/D high byte & status
11101011	235	BSR command - DIM , house 2	A/D high byte & status
11101100	236	BSR address - unit 1, house 2	A/D high byte & status
11101101	237	BSR address - unit 1, house 2	A/D high byte & status
11101110	238	BSR address - unit 9, house 2	A/D high byte & status
11101111	239	BSR address - unit 9, house 2	A/D high byte & status

<u>COMMAND</u> <u>Binary</u>	<u>BYTE VALUE</u> <u>Decimal</u>	<u>ADC-1 ACTION</u> <u>(initiated)</u>	<u>ADC-1 RESPONSE BYTE</u> <u>(contents)</u>
11110000	240	BSR address - unit 14, house 2	meaningless drive1
11110001	241	BSR address - unit 14, house 2	meaningless drive1
11110010	242	BSR address - unit 6, house 2	meaningless drive1
11110011	243	BSR address - unit 6, house 2	meaningless drive1
11110100	244	BSR address - unit 4, house 2	meaningless drive1
11110101	245	BSR address - unit 4, house 2	meaningless drive1
11110110	246	BSR address - unit 12, house 2	meaningless drive1
11110111	247	BSR address - unit 12, house 2	meaningless drive1
11111000	248	BSR address - unit 16, house 2	meaningless drive1
11111001	249	BSR address - unit 16, house 2	meaningless drive1
11111010	250	BSR address - unit 8, house 2	meaningless drive1
11111011	251	BSR address - unit 8, house 2	meaningless drive1
11111100	252	BSR address - unit 2, house 2	meaningless drive1
11111101	253	BSR address - unit 2, house 2	meaningless drive1
11111110	254	BSR address - unit 10, house 2	meaningless drive1
11111111	255	BSR address - unit 10, house 2	meaningless drive1

where house 1 and house 2 refer to the first and second house codes available from one setting of the ADC-1 switches. See Table C-3 for house code details.

C-3 - BSR HOUSE CODE SWITCH SETTINGS

The switches controlling the house code addressed by the ADC-1 may be found on the bottom of the ADC-1. Within the cutout closest to the edge of the enclosure, a bank of eight switches is visible. The switches can be subdivided into 4 pairs of which 3 pairs define the 3 bits needed to specify the 8 house codes pairs. Only one switch within a pair can be on at a given time. When the odd numbered switch is ON and the even numbered switch is OFF, then the pair is OFF. Switches 7 and 8 are not used.

The follow chart can be used to set the house code on the ADC-1 for your application.

House Code	Switch Number					
Pair	1	2	3	4	5	6
A,I	on	off	off	on	off	on
B,J	off	on	off	on	off	on
C,K	on	off	on	off	off	on
D,L	off	on	on	off	off	on
M,E	on	off	on	off	on	off
N,F	off	on	on	off	on	off
O,G	on	off	off	on	on	off
P,H	off	on	off	on	on	off

BSR units using the second house code for each switch setting can be addressed by sending different address and command codes. These codes are listed in Appendix C-4. Within the command sent to the ADC-1, bit 5 is off (0) for the first house code, and is set (1) for the second.

NOTE: The control command codes are different for each of the two houses within a house pair.

C-4 - BSR AC MODULE CONTROL CODES

	<u>Unit</u>	<u>Command/Number</u>	<u>Command</u>	<u>Code</u>
Commands and addresses for first house of house code pair. (A,B,C,D,M,N,O,P)		CLEAR		193
		ALL		195
		ON		197
		OFF		199
		BRIGHT		201
		DIM		203
		1		204
		2		220
		3		196
		4		212
		5		194
		6		210
		7		202
		8		218
		9		206
		10		222
		11		198
		12		214
		13		192
		14		208
		15		200
		16		216
Commands and addresses for second house of house code pair. (I,J,K,L,E,F,G,H)		CLEAR		225
		ALL		227
		ON		229
		OFF		231
		BRIGHT		233
		DIM		235
		1		236
		2		252
		3		228
		4		244
		5		226
		6		242
		7		234
		8		250
		9		238
		10		254
		11		230
		12		246
		13		224
		14		240
		15		232
		16		248

APPENDIX D

DATASHEETS

This appendix contains technical datasheets and application notes for some of the components and sensors of the ADC-1. The datasheets included here are not exact replicas of the original manufacturers datasheets, but have been edited to display the aspects relevant to ADC-1 operation. The full datasheets are available from the original manufacturer as stated on each sheet.

ICL7109 12 Bit Binary A/D Converter for Microprocessor Interfaces

FEATURES

- 12 bit binary (plus polarity and overrange) dual slope integrating analog-to-digital converter.
- Byte-organized TTL-compatible three-state outputs and UART handshake mode for simple parallel or serial interfacing to microprocessor systems.
- RUN/HOLD input and STATUS output can be used to monitor and control conversion timing.
- True differential input and differential reference.
- Low noise — typically $15\mu\text{V}$ p-p.
- 1pA typical input current.
- Operates at up to 30 conversions per second.
- On-chip oscillator operates with inexpensive 3.58MHz TV crystal giving 7.5 conversions per second for 60Hz rejection. May also be operated as RC oscillator for other clock frequencies.
- Fabricated using MAX-CMOS™ technology combining analog and digital functions on a single low power LSI CMOS chip.
- All inputs fully protected against static discharge; no special handling precautions necessary.

GENERAL DESCRIPTION

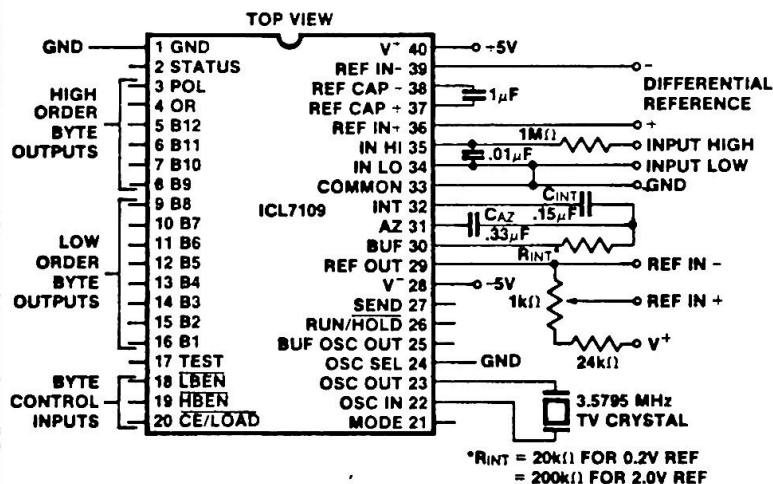
The ICL7109 is a high performance, low power integrating A/D converter designed to easily interface with microprocessors.

The output data (12 bits, polarity and overrange) may be directly accessed under control of two byte enable inputs and a chip select input for a simple parallel bus interface. A UART handshake mode is provided to allow the ICL7109 to work with industry-standard UARTs in providing serial data transmission, ideal for remote data logging applications. The RUN/HOLD input and STATUS output allow monitoring and control of conversion timing.

The ICL7109 provides the user with the high accuracy, low noise, low drift, versatility and economy of the dual-slope integrating A/D converter. Features like true differential input and reference, drift of less than $1\mu\text{V}/^\circ\text{C}$, maximum input bias current of 10pA , and typical power consumption of 20mW make the ICL7109 an attractive per-channel alternative to analog multiplexing for many data acquisition applications.

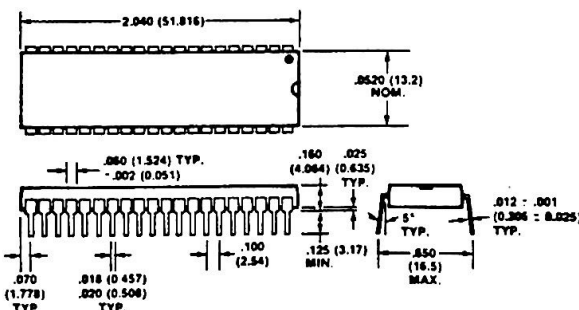
PIN CONFIGURATION AND TEST CIRCUIT:

(See Figure 1 for typical connection to a UART or Microcomputer)

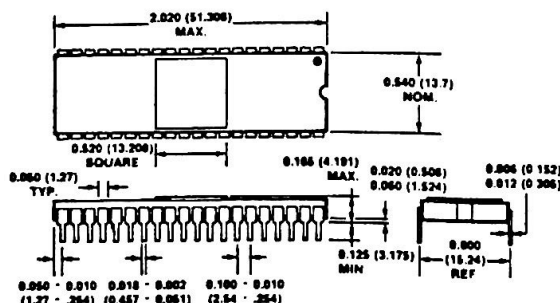


PACKAGE DIMENSIONS

40 Pin Plastic Dual-In-Line Package



40 Pin Ceramic Dual-In-Line Package



ORDERING INFORMATION

PART	PACKAGE	TEMP. RANGE	ORDER PART #
7109	40 pin ceramic DIP	-55°C to +125°C	ICL7109MDL
7109	40 pin ceramic DIP	-25°C to +85°C	ICL7109IDL
7109	40 pin plastic DIP	0°C to +70°C	ICL7109CPL

TABLE 2 - Pin Assignment and Function Description

PIN	SYMBOL	DESCRIPTION
1	GND	Digital Ground, 0V, Ground return for all digital logic
2	STATUS	Output High during integrate and deintegrate until data is latched. Output Low when analog section is in Auto-Zero configuration.
3	POL	Polarity - HI for Positive Input.
4	OR	Overrange - HI if Overranged.
5	B12	Bit 12 (Most Significant Bit)
6	B11	Bit 11
7	B10	Bit 10
8	B9	Bit 9
9	B8	Bit 8
10	B7	Bit 7
11	B6	Bit 6
12	B5	Bit 5
13	B4	Bit 4
14	B3	Bit 3
15	B2	Bit 2
16	B1	Bit 1 (Least Significant Bit)
17	TEST	Input High - Normal Operation. Input Low - Forces all bit outputs high. Note: This input is used for test purposes only.
18	LBEN	Low Byte Enable - With Mode (Pin 21) low, and $\overline{\text{CE/LOAD}}$ (Pin 20) low, taking this pin low activates low order byte outputs B1-B8. - With Mode (Pin 21) high, this pin serves as a low byte flag output used in handshake mode. See Figures 7, 8, 9.
19	HBEN	High Byte Enable - With Mode (Pin 21) low, and $\overline{\text{CE/LOAD}}$ (Pin 20) low, taking this pin low activates high order byte outputs B9-B12, POL, OR. - With Mode (Pin 21) high, this pin serves as a high byte flag output used in handshake mode. See Figures 7, 8, 9.
20	$\overline{\text{CE/LOAD}}$	Chip Enable Load - With Mode (Pin 21) low, $\overline{\text{CE/LOAD}}$ serves as a master output enable. When high, B1-B12, POL, OR outputs are disabled. - With Mode (Pin 21) high, this pin serves as a load strobe used in handshake mode. See Figures 7, 8, 9.

All three state output data bits

PIN	SYMBOL	DESCRIPTION
21	MODE	Input Low - Direct output mode where $\overline{\text{CE/LOAD}}$ (Pin 20), $\overline{\text{HBEN}}$ (Pin 19) and $\overline{\text{LBEN}}$ (Pin 18) act as inputs directly controlling byte outputs. Input Pulsed High - Causes immediate entry into handshake mode and output of data as in Figure 9. Input High - Enables $\overline{\text{CE/LOAD}}$ (Pin 20), $\overline{\text{HBEN}}$ (Pin 19), and $\overline{\text{LBEN}}$ (Pin 18) as outputs, handshake mode will be entered and data output as in Figures 7 and 8 at conversion completion.
22	OSC IN	Oscillator Input
23	OSC OUT	Oscillator Output
24	OSC SEL	Oscillator Select - Input high configures OSC IN, OSC OUT, BUF OSC OUT as RC oscillator - clock will be same phase and duty cycle as BUF OSC OUT. - Input low configures OSC IN, OSC OUT for crystal oscillator - clock frequency will be 1/58 of frequency at BUF OSC OUT.
25	BUF OSC OUT	Buffered Oscillator Output
26	RUN/HOLD	Input High - Conversions continuously performed every 8192 clock pulses. Input Low - Conversion in progress completed, converter will stop in Auto-Zero 7 counts before integrate.
27	SEND	Input - Used in handshake mode to indicate ability of an external device to accept data.
28	V ⁻	Analog Negative Supply - Nominally -5V with respect to GND (Pin 1).
29	REF OUT	Reference Voltage Output - Nominally 2.8V down from V ⁺ (Pin 40).
30	BUFFER	Buffer Amplifier Output
31	AUTO-ZERO	Auto-Zero Node - Inside foil of C _{AZ}
32	INTEGRATOR	Integrator Output - Outside foil of C _{INT}
33	COMMON	Analog Common - System is Auto-Zeroed to COMMON
34	INPUT LO	Differential Input Low Side
35	INPUT HI	Differential Input High Side
36	REF IN +	Differential Reference Input Positive
37	REF CAP +	Reference Capacitor Positive
38	REF CAP -	Reference Capacitor Negative
39	REF IN -	Differential Reference Input Negative
40	V ⁺	Positive Supply Voltage - Nominally +5V with respect to GND (Pin 1).

Note: All digital levels are positive true.

DETAILED DESCRIPTION

Analog Section

Figure 2 shows the equivalent circuit of the Analog Section of the ICL7109. When the RUN/HOLD input is left open or connected to V⁺, the circuit will perform conversions at a rate determined by the clock frequency (8192 clock periods per cycle). Each measurement cycle is divided into three phases as shown in Figure 3. They are (1) Auto-Zero (AZ), (2) Signal Integrate (INT) and (3) Deintegrate (DE).

1. Auto-Zero Phase

During auto-zero three things happen. First, input high and low are disconnected from their pins and internally shorted to analog common. Second, the reference capacitor is charged to the reference voltage. Third, a feedback loop is closed around the system to charge the auto-

zero capacitor C_{AZ} to compensate for offset voltages in the buffer amplifier, integrator, and comparator. Since the comparator is included in the loop, the AZ accuracy is limited only by the noise of the system. In any case, the offset referred to the input is less than 10 μ V.

2. Signal Integrate Phase

During signal integrate the auto-zero loop is opened, the internal short is removed and the internal input high and low are connected to the external pins. The converter then integrates the differential voltage between input high and input low for a fixed time of 2048 clock periods. At the end of this phase, the polarity of the integrated signal is determined.

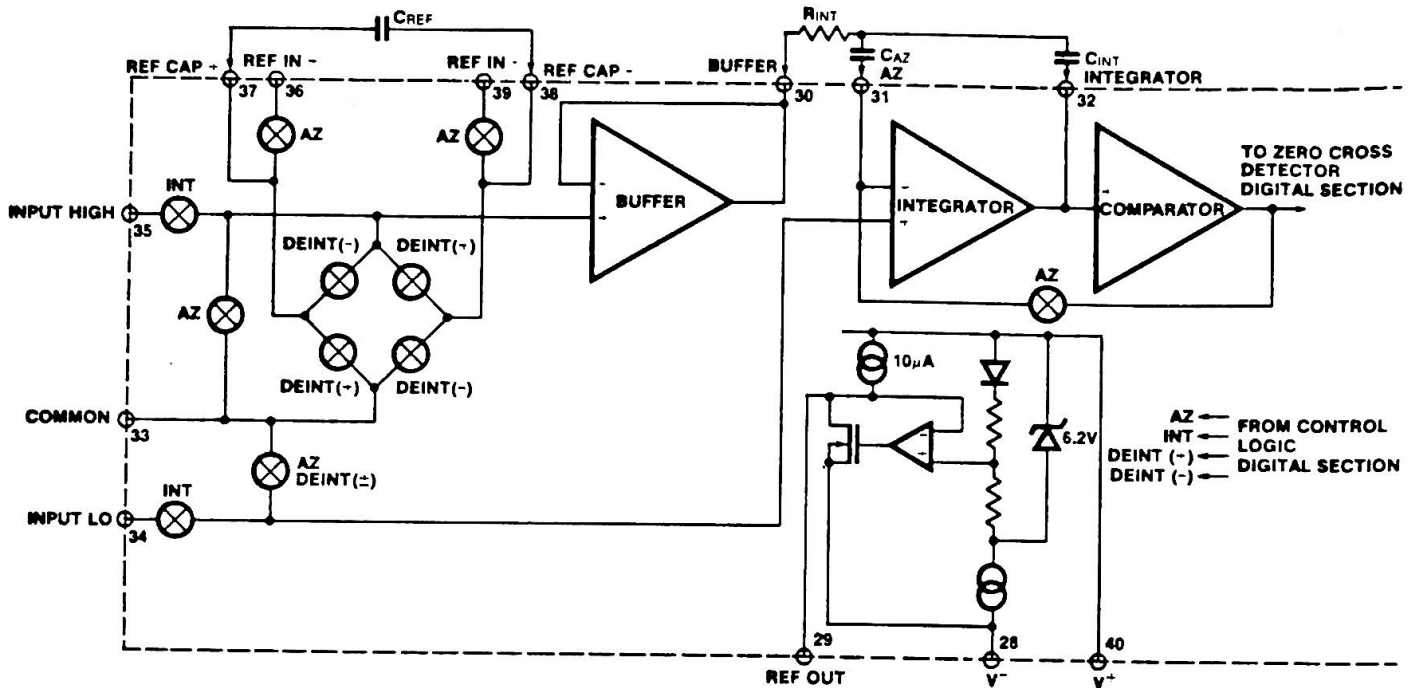


Figure 2: Analog Section

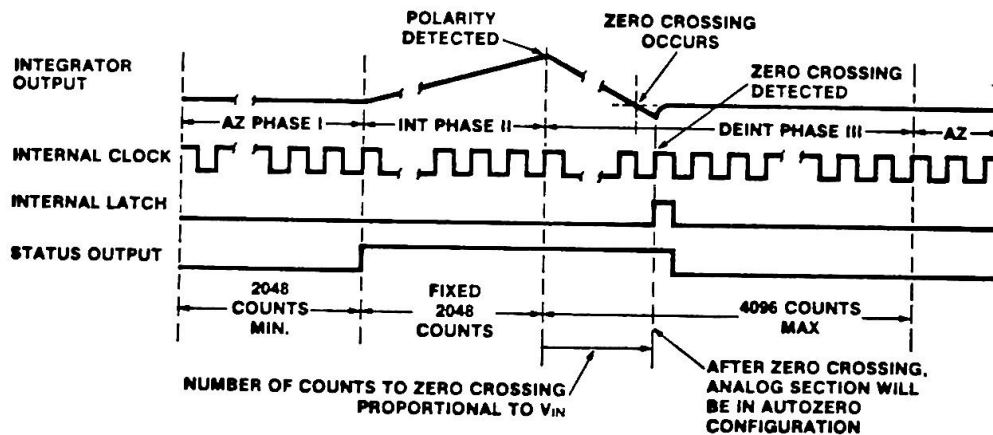


Figure 3: Conversion Timing (RUN/HOLD Pin High)

3. Deintegrate Phase

The final phase is deintegrate, or reference integrate. Input low is internally connected to analog common and input high is connected across the previously charged (during auto-zero) reference capacitor. Circuitry within the chip ensures that the capacitor will be connected with the correct polarity to cause the integrator output to return to the zero crossing (established in Auto Zero) with a fixed slope. Thus the time for the output to return to zero (represented by the number of clock periods counted) is proportional to the input signal.

Differential Input

The input can accept differential voltages anywhere within the common mode range of the input amplifier; or specifically from 0.5 volts below the positive supply to 1.0 volt above the negative supply. In this range the system has a CMRR of 86dB typical. However, since the integrator also swings with the common mode voltage, care must be exercised to assure the integrator output does not saturate. A worst case condition would be a large positive common mode voltage with a near full-scale negative differential input voltage. The negative input signal drives the integrator

positive when most of its swing has been used up by the positive common mode voltage. For these critical applications the integrator swing can be reduced to less than the recommended 4V full scale with some loss of accuracy. The integrator output can swing within 0.3 volts of either supply without loss of linearity.

The ICL7109 has, however, been optimized for operation with analog common near digital ground. With power supplies of +5V and -5V, this allows a 4V full scale integrator swing positive or negative maximizing the performance of the analog section.

Differential Reference

The reference voltage can be generated anywhere within the power supply voltage of the converter. The main source of common mode error is a roll-over voltage caused by the reference capacitor losing or gaining charge to stray capacity on its nodes. If there is a large common mode voltage, the reference capacitor can gain charge (increase voltage) when called up to deintegrate a positive signal but lose charge (decrease voltage) when called up to deintegrate a negative input signal. This difference in reference for (+) or (-) input voltage will give a roll-over error. However, by

selecting the reference capacitor large enough in comparison to the stray capacitance, this error can be held to less than 0.5 count for the worst case condition (see Component Values Selection below).

The roll-over error from these sources is minimized by having the reference common mode voltage near or at analog common.

Component Value Selection

For optimum performance of the analog section, care must be taken in the selection of values for the integrator capacitor and resistor, auto-zero capacitor, reference voltage, and conversion rate. These values must be chosen to suit the particular application.

The most important consideration is that the integrator output swing (for full-scale input) be as large as possible. For example, with $\pm 5V$ supplies and COMMON connected to GND, the nominal integrator output swing at full scale is $\pm 4V$. Since the integrator output can go to 0.3V from either supply without significantly affecting linearity, a 4V integrator output swing allows 0.7V for variations in output swing due to component value and oscillator tolerances. With $\pm 5V$ supplies and a common mode range of $\pm 1V$ required, the component values should be selected to provide $\pm 3V$ integrator output swing. Noise and rollover errors will be slightly worse than in the $\pm 4V$ case. For larger common mode voltage ranges, the integrator output swing must be reduced further. This will increase both noise and rollover errors. To improve the performance, supplies of $\pm 6V$ may be used.

1. Integrating Resistor

Both the buffer amplifier and the integrator have a class A output stage with $100\mu A$ of quiescent current. They supply $20\mu A$ of drive current with negligible non-linearity. The integrating resistor should be large enough to remain in this very linear region over the input voltage range, but small enough that undue leakage requirements are not placed on the PC board. For 4.096 volt full scale, $200k\Omega$ is near optimum and similarly a $20k\Omega$ for a 409.6mV scale. For other values of full scale voltage, R_{INT} should be chosen by the relation

$$R_{INT} = \frac{\text{full scale voltage}}{20\mu A}$$

2. Integrating Capacitor

The integrating capacitor C_{INT} should be selected to give the maximum integrator output voltage swing without saturating the integrator (approximately 0.3 volt from either supply). For the ICL7109 with ± 5 volt supplies and analog common connected to GND, a ± 3.5 to ± 4 volt integrator output swing is nominal. For 7-1/2 conversions per second (61.72KHz clock frequency) as provided by the crystal oscillator, nominal values for C_{INT} and C_{AZ} are $0.15\mu F$ and $0.33\mu F$, respectively. If different clock frequencies are used, these values should be changed to maintain the integrator output voltage swing. In general, the value of C_{INT} is given by

$$C_{INT} = \frac{(2048 \times \text{clock period}) (20\mu)}{\text{integrator output voltage swing}}$$

An additional requirement of the integrating capacitor is that it have low dielectric absorption to prevent roll-over errors. While other types of capacitors are adequate for this application, polypropylene capacitors give undetectable errors at reasonable cost up to $85^\circ C$. For the military temperature range, Teflon® capacitors are recommen-

ded. While their dielectric absorption characteristics vary somewhat from unit to unit, selected devices should give less than 0.5 count of error due to dielectric absorption.

3. Auto-Zero Capacitor

The size of the auto-zero capacitor has some influence on the noise of the system; a big capacitor, giving less noise. However, it cannot be increased without limits since it, in parallel with the integrating capacitor forms an R-C time constant that determines the speed of recovery from overloads and more important the error that exists at the end of an auto-zero cycle. For 409.6mV full scale where noise is very important and the integrating resistor small, a value of C_{AZ} twice C_{INT} is optimum. Similarly for 4.096V full scale where recovery is more important than noise, a value of C_{AZ} equal to half of C_{INT} is recommended.

For optimal rejection of stray pickup, the outer foil of C_{AZ} should be connected to the R-C summing junction and the inner foil to pin 31. Similarly the outer foil of C_{INT} should be connected to pin 32 and the inner foil to the R-C summing junction. Teflon®, or equivalent, capacitors are recommended above $85^\circ C$ for their low leakage characteristics.

4. Reference Capacitor

A $1\mu F$ capacitor gives good results in most applications. However, where a large common mode voltage exists (i.e. the reference low is not at analog common) and a 409.6mV scale is used, a larger value is required to prevent roll-over error. Generally $10\mu F$ will hold the roll-over error to 0.5 count in this instance. Again, Teflon®, or equivalent capacitors should be used for temperatures above $85^\circ C$ for their low leakage characteristics.

5. Reference Voltage

The analog input required to generate a full scale output of 4096 counts is $V_{IN} = 2V_{REF}$. Thus for a normalized scale, a reference of 2.048V should be used for a 4.096V full scale, and 204.8mV should be used for a 0.4096V full scale. However, in many applications where the A/D is sensing the output of a transducer, there will exist a scale factor other than unity between the absolute output voltage to be measured and a desired digital output. For instance, in a weighing system, the designer might like to have a full scale reading when the voltage from the transducer is 0.682V. Instead of dividing the input down to 409.6mV, the input voltage should be measured directly and a reference voltage of 0.341V should be used. Suitable values for integrating resistor and capacitor are $34k$ and $0.15\mu F$. This avoids a divider on the input. Another advantage of this system occurs when a zero reading is desired for non-zero input. Temperature and weight measurements with an offset or tare are examples. The offset may be introduced by connecting the voltage output of the transducer between common and analog high, and the offset voltage between common and analog low, observing polarities carefully. However, in processor-based systems using the ICL7109, it may be more efficient to perform this type of scaling or tare subtraction digitally using software.

6. Reference Sources

The stability of the reference voltage is a major factor in the overall absolute accuracy of the converter. The resolution of the ICL7109 at 12 bits is one part in 4096, or 244ppm. Thus if the reference has a temperature coefficient of 80ppm/ $^\circ C$ (onboard reference) a temperature difference of $3^\circ C$ will introduce a one-bit absolute error.

INTERSIL

IM6402/IM6403 Universal Asynchronous Receiver Transmitter (UART)

FEATURES

- Low Power — Less Than 10mW Typ. at 2MHz
- Operation Up to 4MHz Clock (IM6402A)
- Programmable Word Length, Stop Bits and Parity
- Automatic Data Formatting and Status Generation
- Compatible with Industry Standard UART's (IM6402)
- On-Chip Oscillator with External Crystal (IM6403)
- Operating Voltage —
IM6402-1/03-1: 5V
IM6402A/03A: 4-11V
IM6402/03: 5V

GENERAL DESCRIPTION

The IM6402 and IM6403 are CMOS/LSI UART's for interfacing computers or microprocessors to asynchronous serial data channels. The receiver converts serial start, data, parity and stop bits to parallel data verifying proper code transmission, parity, and stop bits. The transmitter converts parallel data into serial form and automatically adds start, parity, and stop bits.

The data word length can be 5, 6, 7 or 8 bits. Parity may be odd or even, and parity checking and generation can be inhibited. The stop bits may be one or two (or one and one-half when transmitting 5 bit code). Serial data format is shown in Figure 6.

The IM6402 and IM6403 can be used in a wide range of applications including modems, printers, peripherals and remote data acquisition systems. CMOS/LSI technology permits clock frequencies up to 4.0MHz (250K Baud), an improvement of 10 to 1 over previous PMOS UART designs. Power requirements, by comparison, are reduced from 670mW to 10mW. Status logic increases flexibility and simplifies the user interface.

The IM6402 differs from the IM6403 in the use of five device pins as indicated in Table 1 and Figure 1.

PIN CONFIGURATION (outline dwg DL, PL)

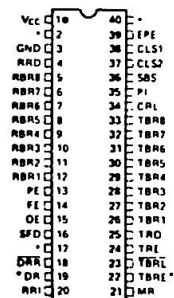


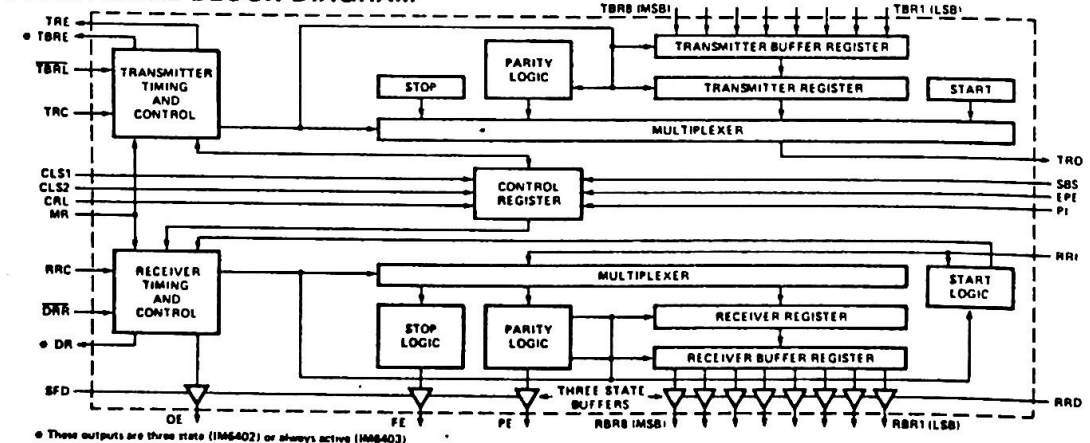
TABLE 1

PIN	IM6402	IM6403 w/XTAL	IM6403 w/EXT CLOCK
2	N/C	Divide Control	Divide Control
17	RRC	XTAL	External Clock Input
19	Tri State	Always Active	Always Active
22	Tri State	Always Active	Always Active
40	TRC	XTAL	GND

ORDERING INFORMATION

ORDER CODE	IM6402-1/03-1	IM6402A/03A	IM6402/03
PLASTIC PKG	IM6402-1/03-1IPL	IM6402/03-AIPL	IM6402/03-IPL
CERAMIC PKG	IM6402-1/03-1IDL	IM6402/03-AIDL	IM6402/03-IDL
MILITARY TEMP	IM6402-1/03-1MDL	IM6402/03-AMDL	—
MILITARY TEMP. WITH 863B	IM6402-1/03-1MDL/863B	IM6402/03-AMD/863B	—

FUNCTIONAL BLOCK DIAGRAM



INTERSIL

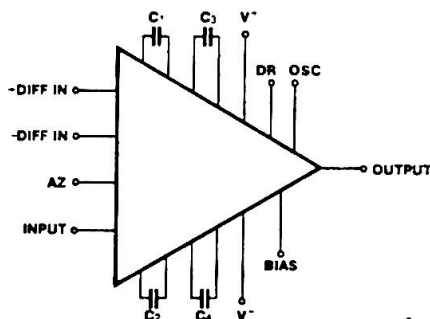
ICL7605/ICL7606

Commutating Auto-Zero (CAZ) Instrumentation Amplifier

FEATURES

- Exceptionally low input offset voltage — $2\mu\text{V}$
- Low long term input offset voltage drift — $0.2\mu\text{V}/\text{year}$
- Low input offset voltage temperature drift — $0.05\mu\text{V}/^\circ\text{C}$
- Wide common mode input voltage range — 0.3V above supply rail
- High common mode rejection ratio — 100 dB
- Operates at supply voltages as low as $\pm 2\text{V}$
- Short circuit protection on outputs for $\pm 5\text{V}$ operation
- Static-protected inputs — no special handling required
- Fabricated using proprietary MAXCMOS™ process technology
- Compensated (ICL7605) or uncompensated (ICL7606) versions

SYMBOL



GENERAL DESCRIPTION

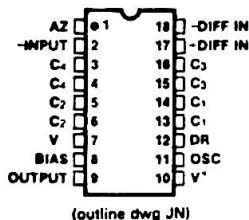
The ICL7605/ICL7606 commutating auto-zero (CAZ) instrumentation amplifiers are designed to replace most of today's hybrid or monolithic instrumentation amplifiers, for low frequency applications from DC to 10 Hz . This is made possible by the unique construction of this new Intersil device, which takes an entirely new design approach to low frequency amplifiers.

Unlike conventional amplifier designs, which employ three op amps and require ultra-high accuracy in resistor tracking and matching, the CAZ instrumentation amplifier requires no trimming except for gain. The key features of the CAZ principle involve automatic compensation for long term drift phenomena and temperature effects, and a flying capacitor input.

The ICL7605/ICL7606 is a monolithic CMOS chip which consists of two analog sections — a unity gain differential to single-ended voltage converter and a CAZ op amp. The first section senses the differential input and applies it to the CAZ amp section. This section consists of an operational amplifier circuit which continuously corrects itself for input voltage errors, such as input offset voltage, temperature effects, and long term drift.

The ICL7605/ICL7606 is intended for low-frequency operation in applications such as strain gauges, which require voltage gains from 1 to 1000 and bandwidths from DC to 10 Hz . Since the CAZ amp automatically corrects itself for internal errors, the only periodic adjustment required is that of gain, which is established by two external resistors. The no-adjustment feature, combined with extremely low offset and temperature coefficient figures, makes the CAZ instrumentation amplifier very desirable for operation in severe environments (temperature, humidity, toxicity, radiation, etc.) where equipment service is difficult.

PIN CONFIGURATION



ORDERING INFORMATION

Order parts by the following part numbers:

Compensated	Uncompensated	Package	Temperature Range
ICL7605CJN	ICL7606CJN	CERDIP	0°C to -70°C
ICL7605IJN	ICL7606IJN	CERDIP	-25°C to $+85^\circ\text{C}$
ICL7605MJN	ICL7606MJN	CERDIP	-55°C to $+125^\circ\text{C}$

Order dice by the following part numbers:

ICL7605/D
ICL7606/D



Two-Terminal IC Temperature Transducer

AD590*

FEATURES

Linear Current Output: $1\mu\text{A}/^\circ\text{K}$
Wide Range: -55°C to $+150^\circ\text{C}$
Probe Compatible Ceramic Sensor Package
Two-Terminal Device: Voltage In/Current Out
Laser Trimmed to $\pm 0.5^\circ\text{C}$ Calibration Accuracy (AD590M)
Excellent Linearity: $\pm 0.3^\circ\text{C}$ Over Full Range (AD590M)
Wide Power Supply Range: $+4\text{V}$ to $+30\text{V}$
Sensor Isolation from Case
Low Cost

PRODUCT DESCRIPTION

The AD590 is a two-terminal integrated circuit temperature transducer which produces an output current proportional to absolute temperature. For supply voltages between $+4\text{V}$ and $+30\text{V}$ the device acts as a high impedance, constant current regulator passing $1\mu\text{A}/^\circ\text{K}$. Laser trimming of the chip's thin film resistors is used to calibrate the device to $298.2\mu\text{A}$ output at 298.2°K ($+25^\circ\text{C}$).

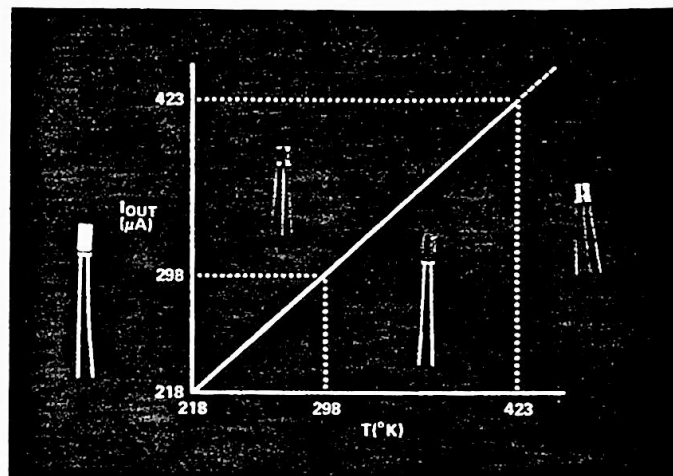
The AD590 should be used in any temperature sensing application below $+150^\circ\text{C}$ in which conventional electrical temperature sensors are currently employed. The inherent low cost of a monolithic integrated circuit combined with the elimination of support circuitry makes the AD590 an attractive alternative for many temperature measurement situations. Linearization circuitry, precision voltage amplifiers, resistance measuring circuitry and cold junction compensation are not needed in applying the AD590.

In addition to temperature measurement, applications include temperature compensation or correction of discrete components, biasing proportional to absolute temperature, flow rate measurement, level detection of fluids and anemometry. The AD590 is available in chip form making it suitable for hybrid circuits and fast temperature measurements in protected environments.

The AD590 is particularly useful in remote sensing applications. The device is insensitive to voltage drops over long lines due to its high impedance current output. Any well-insulated twisted pair is sufficient for operation hundreds of feet from the receiving circuitry. The output characteristics also make the AD590 easy to multiplex: the current can be switched by a CMOS multiplexer or the supply voltage can be switched by a logic gate output.

*Covered by Patent No. 4,123,698

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use; nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices.



PRODUCT HIGHLIGHTS

1. The AD590 is a calibrated two terminal temperature sensor requiring only a dc voltage supply ($+4\text{V}$ to $+30\text{V}$). Costly transmitters, filters, lead wire compensation and linearization circuits are all unnecessary in applying the device.
2. State-of-the-art laser trimming at the wafer level in conjunction with extensive final testing insures that AD590 units are easily interchangeable.
3. Superior interference rejection results from the output being a current rather than a voltage. In addition, power requirements are low (1.5mW 's @ 5V @ $+25^\circ\text{C}$). These features make the AD590 easy to apply as a remote sensor.
4. The high output impedance ($>10\text{M}\Omega$) provides excellent rejection of supply voltage drift and ripple. For instance, changing the power supply from 5V to 10V results in only a $1\mu\text{A}$ maximum current change, or 1°C equivalent error.
5. The AD590 is electrically durable: it will withstand a forward voltage up to 44V and a reverse voltage of 20V . Hence, supply irregularities or pin reversal will not damage the device.
6. The device is hermetically sealed in both a ceramic sensor package and in to TO-52 package. MIL-STD-883 processing to level B is available and, for large unit volumes, special accuracy requirements over limited temperature ranges can be satisfied by selections at final test. The device is also available in chip form.

P.O. Box 280; Norwood, Massachusetts 02062 U.S.A.
Tel: 617/329-4700
Telex: 924491
Twx: 710/394-6577
Cables: ANALOG NORWOODMASS

SPECIFICATIONS (typical @ +25°C and $V_S = 5V$ unless otherwise noted)

MODEL	AD590I	AD590J	AD590K	AD590L	AD590M
ABSOLUTE MAXIMUM RATINGS					
Forward Voltage (E+ to E-)	+44V	•	•	•	•
Reverse Voltage (E+ to E-)	-20V	•	•	•	•
Breakdown Voltage (Case to E+ or E-)	±200V	•	•	•	•
Rated Performance Temperature Range ¹	-55°C to +150°C	•	•	•	•
Storage Temperature Range ¹	-65°C to +155°C	•	•	•	•
Lead Temperature (Soldering, 10 sec)	+300°C	•	•	•	•
POWER SUPPLY					
Operating Voltage Range	+4V to +30V	•	•	•	•
OUTPUT					
Nominal Current Output @ +25°C (298.2°K)	298.2μA	•	•	•	•
Nominal Temperature Coefficient	1μA/°K	•	•	•	•
Calibration Error @ +25°C	±10.0°C max	±5.0°C max	±2.5°C max	±1.0°C max	±0.5°C max
Absolute Error ² (over rated performance temperature range)					
Without External Calibration Adjustment	±20.0°C max	±10.0°C max	±5.5°C max	±3.0°C max	±1.7°C max
With +25°C Calibration Error Set to Zero	±5.8°C max	±3.0°C max	±2.0°C max	±1.6°C max	±1.0°C max
Nonlinearity	±3.0°C max	±1.5°C max	±0.8°C max	±0.4°C max	±0.3°C max
Repeatability ³	±0.1°C max	•	•	•	•
Long Term Drift ⁴	±0.1°C/month max	•	•	•	•
Current Noise	40pA/√Hz	•	•	•	•
Power Supply Rejection					
+4V < V_S < +5V	0.5μA/V	•	•	•	•
+5V < V_S < +15V	0.2μA/V	•	•	•	•
+15V < V_S < +30V	0.1μA/V	•	•	•	•
Case Isolation to Either Lead	10 ¹⁰ Ω	•	•	•	•
Effective Shunt Capacitance	100pF	•	•	•	•
Electrical Turn-On Time ⁵	20μs	•	•	•	•
Reverse Bias Leakage Current ⁶ (Reverse Voltage = 10V)	10pA	•	•	•	•

NOTES:

*Specifications same as AD590I

¹The AD590 has been used at -100°C and +200°C for short periods of measurement with no physical damage to the device. However, the absolute errors specified apply to only the rated performance temperature range.

²See page 8-10 for explanation of error components. Note that ±1°C error is the equivalent of ±1μA error.

³Maximum deviation between +25°C readings after temperature cycling between -55°C and +150°C; guaranteed not tested.

⁴Conditions: constant +5V, constant +125°C; guaranteed, not tested.

⁵Does not include self heating effects; see page 8-11 for explanation of these effects.

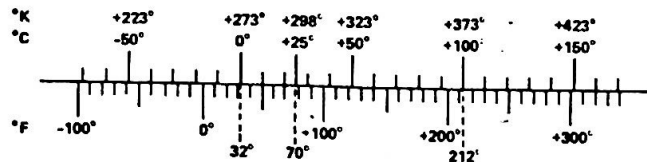
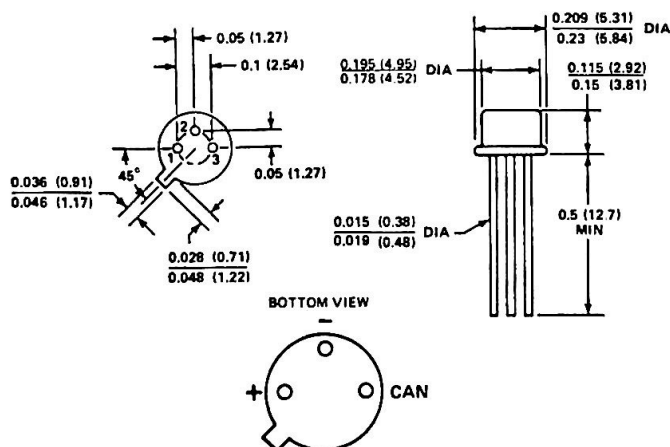
⁶Leakage current doubles every 10°C.

Specifications subject to change without notice.

OUTLINE DIMENSIONS AND PIN DESIGNATIONS

Dimensions shown in inches and (mm).

TO-52 PACKAGE: DESIGNATION "H"



TEMPERATURE SCALE CONVERSION EQUATIONS

$$^{\circ}\text{C} = \frac{5}{9} (^{\circ}\text{F} - 32) \quad ^{\circ}\text{K} = ^{\circ}\text{C} + 273.15$$

$$^{\circ}\text{F} = \frac{9}{5} ^{\circ}\text{C} + 32 \quad ^{\circ}\text{R} = ^{\circ}\text{F} + 459.7$$

The 590H has 60μ inches of gold plating on its Kovar leads and Kovar header. A resistance welder is used to seal the nickel cap to the header. The AD590 chip is eutectically mounted to the header and ultrasonically bonded to with 1 MIL aluminum wire. Kovar composition: 53% iron nominal; 29% ±1% nickel; 17% ±1% cobalt; 0.65% manganese max; 0.20% silicon max; 0.10% aluminum max; 0.10% magnesium max; 0.10% zirconium max; 0.10% titanium max; 0.06% carbon max.

EXPLANATION OF TEMPERATURE SENSOR SPECIFICATIONS

The way in which the AD590 is specified makes it easy to apply in a wide variety of different applications. It is important to understand the meaning of the various specifications and the effects of supply voltage and thermal environment on accuracy.

The AD590 is basically a PTAT (proportional to absolute temperature)¹ current regulator. That is, the output current is equal to a scale factor times the temperature of the sensor in degrees Kelvin. This scale factor is trimmed to $1\mu\text{A}/^\circ\text{K}$ at the factory, by adjusting the indicated temperature (i.e. the output current) to agree with the actual temperature. This is done with 5V across the device at a temperature within a few degrees of 25°C (298.2°K). The device is then packaged and tested for accuracy over temperature.

CALIBRATION ERROR

At final factory test the difference between the indicated temperature and the actual temperature is called the calibration error. Since this is a scale factor error, its contribution to the total error of the device is PTAT. For example, the effect of the 1°C specified maximum error of the AD590L varies from 0.73°C at -55°C to 1.42°C at 150°C . Figure 3 shows how an exaggerated calibration error would vary from the ideal over temperature.

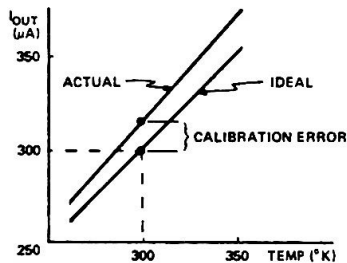


Figure 3. Calibration Error vs. Temperature

The calibration error is a primary contributor to maximum total error in all AD590 grades. However, since it is a scale factor error, it is particularly easy to trim. Figure 4 shows the most elementary way of accomplishing this. To trim this circuit the temperature of the AD590 is measured by a reference temperature sensor and R is trimmed so that $V_T = 1\text{mV}/^\circ\text{K}$ at that temperature. Note that when this error is trimmed out at one temperature, its effect is zero over the entire temperature range. In most applications there is a current to voltage conversion resistor (or, as with a current input ADC, a reference) that can be trimmed for scale factor adjustment.

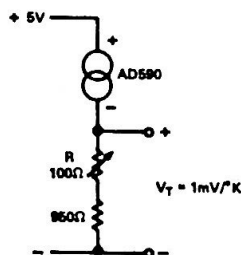


Figure 4. One Temperature Trim

¹ $T(^{\circ}\text{C}) = T(^{\circ}\text{K}) - 273.2$; Zero on the Kelvin scale is "absolute zero"; there is no lower temperature.

ERROR VERSUS TEMPERATURE: WITH CALIBRATION ERROR TRIMMED OUT

Each AD590 is also tested for error over the temperature range with the calibration error trimmed out. This specification could also be called the "variance from PTAT" since it is the maximum difference between the actual current over temperature and a PTAT multiplication of the actual current at 25°C . This error consists of a slope error and some curvature, mostly at the temperature extremes. Figure 5 shows a typical AD590K temperature curve before and after calibration error trimming.

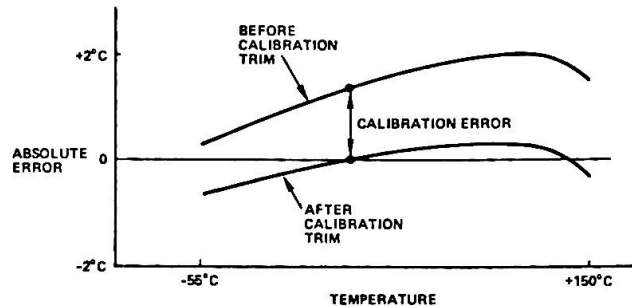


Figure 5. Effect of Scale Factor Trim on Accuracy

ERROR VERSUS TEMPERATURE: NO USER TRIMS

Using the AD590 by simply measuring the current, the total error is the "variance from PTAT" described above plus the effect of the calibration error over temperature. For example the AD590L maximum total error varies from 2.33°C at -55°C to 3.02°C at 150°C . For simplicity, only the larger figure is shown on the specification page.

NONLINEARITY

Nonlinearity as it applies to the AD590 is the maximum deviation of current over temperature from a best-fit straight line. The nonlinearity of the AD590 over the -55°C to $+150^\circ\text{C}$ range is superior to all conventional electrical temperature sensors such as thermocouples, RTD's and thermistors. Figure 6 shows the nonlinearity of the typical AD590K from Figure 5.

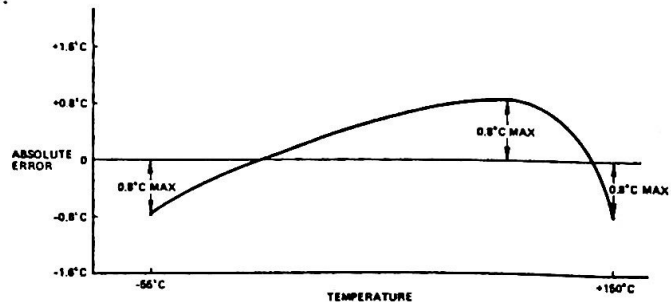


Figure 6. Nonlinearity

Figure 7A shows a circuit in which the nonlinearity is the major contributor to error over temperature. The circuit is trimmed by adjusting R_1 for a 0V output with the AD590 at 0°C . R_2 is then adjusted for 10V out with the sensor at 100°C . Other pairs of temperatures may be used with this procedure as long as they are measured accurately by a reference sensor. Note that for $+15\text{V}$ output (150°C) the $V+$ of the op amp must be greater than 17V . Also note that $V-$ should be at least -4V ; if $V-$ is ground there is no voltage applied across the device.

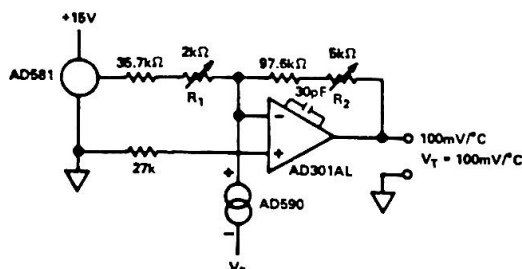


Figure 7A. Two Temperature Trim

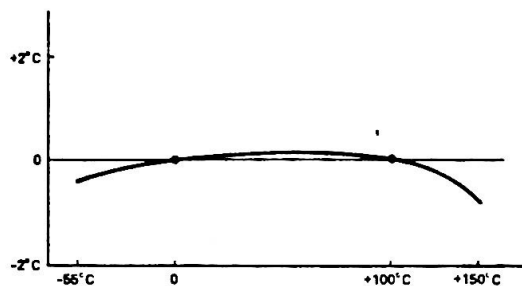


Figure 7B. Typical Two-Trim Accuracy

VOLTAGE AND THERMAL ENVIRONMENT EFFECTS

The power supply rejection specifications show the maximum expected change in output current versus input voltage changes. The insensitivity of the output to input voltage allows the use of unregulated supplies. It also means that hundreds of ohms of resistance (such as a CMOS multiplexer) can be tolerated in series with the device.

It is important to note that using a supply voltage other than 5V does not change the PTAT nature of the AD590. In other words, this change is equivalent to a calibration error and can be removed by the scale factor trim (see previous page).

The AD590 specifications are guaranteed for use in a low thermal resistance environment with 5V across the sensor. Large changes in the thermal resistance of the sensor's environment will change the amount of self-heating and result in changes in the output which are predictable but not necessarily desirable.

The thermal environment in which the AD590 is used determines two important characteristics: the effect of self heating and the response of the sensor with time.

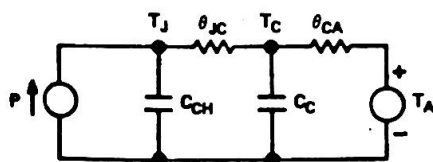


Figure 8. Thermal Circuit Model

Figure 8 is a model of the AD590 which demonstrates these characteristics. As an example, for the TO-52 package, θ_{JC} is the thermal resistance between the chip and the case, about

26°C/watt. θ_{CA} is the thermal resistance between the case and its surroundings and is determined by the characteristics of the thermal connection. Power source P represents the power dissipated on the chip. The rise of the junction temperature, T_J , above the ambient temperature T_A is:

$$T_J - T_A = P (\theta_{JC} + \theta_{CA}). \quad \text{Eq. 1}$$

Table I gives the sum of θ_{JC} and θ_{CA} for several common thermal media for both the "H" and "F" packages. The heat-sink used was a common clip-on. Using Equation 1, the temperature rise of an AD590 "H" package in a stirred bath at +25°C, when driven with a 5V supply, will be 0.06°C. However, for the same conditions in still air the temperature rise is 0.72°C. For a given supply voltage, the temperature rise varies with the current and is PTAT. Therefore, if an application circuit is trimmed with the sensor in the same thermal environment in which it will be used, the scale factor trim compensates for this effect over the entire temperature range.

MEDIUM	$\theta_{JC} + \theta_{CA}$ (°C/watt)		τ (sec) (Note 3)	
	H	F	H	F
Aluminum Block	30	10	0.6	0.1
Stirred Oil ¹	42	60	1.4	0.6
Moving Air ²				
With Heat Sink	45	—	5.0	—
Without Heat Sink	115	190	13.5	10.0
Still Air				
With Heat Sink	191	—	108	—
Without Heat Sink	480	650	60	30

¹ Note: τ is dependent upon velocity of oil; average of several velocities listed above.

² Air velocity $\approx$ 9ft/sec.

³ The time constant is defined as the time required to reach 63.2% of an instantaneous temperature change.

Table I. Thermal Resistances

The time response of the AD590 to a step change in temperature is determined by the thermal resistances and the thermal capacities of the chip, C_{CH} , and the case, C_C . C_{CH} is about 0.04 watt-sec/°C for the AD590. C_C varies with the measured medium since it includes anything that is in direct thermal contact with the case. In most cases, the single time constant exponential curve of Figure 9 is sufficient to describe the time response, $T(t)$. Table 1 shows the effective time constant, τ , for several media.

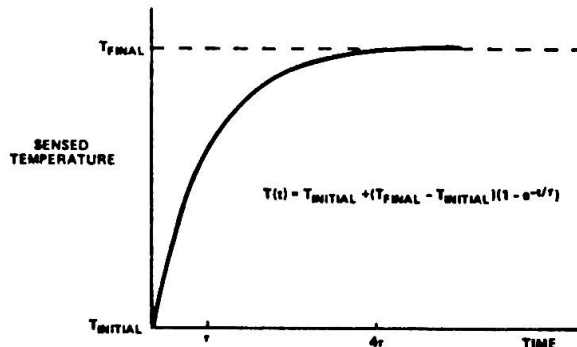


Figure 9. Time Response Curve

ACCURACIES OF THE AD590

The following tables contain maximum errors by grade for applications involving limited temperature spans. The tables reflect the worst-case nonlinearities of the AD590, which invariably occur at the ends of the specified temperature range. The "trims" in each table refer to the error correction circuits on pages 4 and 5 of the AD590 Data Sheet (Figures

4 and 7a). All accuracies given below are $\pm^{\circ}\text{C}$. For example, if $\pm 1^{\circ}\text{C}$ accuracy is required over the $+25^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ range, then trimming a J grade device using the circuit of Figure 4 on the AD590 Data Sheet will result in a sensor of the required accuracy and range.

M GRADE

Number Of Trims	Temperature Span ($^{\circ}\text{C}$)	Lowest Temperature In Span ($^{\circ}\text{C}$)							
		-55	-25	0	+25	+50	+75	+100	+125
None	10	0.6	0.5	0.6	0.6	0.7	0.7	0.7	0.9
None	25	0.8	0.8	0.7	0.7	0.8	0.8	1.0	1.1
None	50	1.0	0.9	0.8	0.9	0.9	1.1	1.2	—
None	100	1.3	1.4	1.3	1.4	1.5	—	—	—
None	150	1.5	1.6	1.6	—	—	—	—	—
None	205	1.7	—	—	—	—	—	—	—
One	10	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.2
One	25	0.4	0.3	0.2	0.2	0.2	0.2	0.3	0.4
One	50	0.5	0.4	0.3	0.3	0.3	0.4	0.5	—
One	100	0.8	0.8	0.7	0.7	0.8	—	—	—
One	150	0.9	0.9	0.9	—	—	—	—	—
One	205	1.0	—	—	—	—	—	—	—
Two	10	0.1	*	*	*	*	*	*	0.1
Two	25	0.1	*	*	*	*	*	*	0.1
Two	50	0.2	*	*	*	*	*	0.2	—
Two	100	0.2	0.1	*	0.1	0.2	—	—	—
Two	150	0.3	0.2	0.3	—	—	—	—	—
Two	205	0.3	—	—	—	—	—	—	—

J GRADE

Number Of Trims	Temperature Span ($^{\circ}\text{C}$)	Lowest Temperature In Span ($^{\circ}\text{C}$)							
		-55	-25	0	+25	+50	+75	+100	+125
None	10	4.2	4.6	5.0	5.4	5.8	6.2	6.6	7.2
None	25	5.0	5.2	5.5	5.9	6.0	6.9	7.5	8.0
None	50	6.5	6.5	6.4	6.9	7.3	8.2	9.0	—
None	100	7.7	8.0	8.3	8.7	9.4	—	—	—
None	150	9.2	9.5	9.6	—	—	—	—	—
None	205	10.0	—	—	—	—	—	—	—
One	10	0.3	0.2	0.2	0.2	0.2	0.2	0.2	0.3
One	25	0.9	0.6	0.5	0.5	0.5	0.6	0.8	0.9
One	50	1.9	1.5	1.0	1.0	1.0	1.5	1.9	—
One	100	2.3	2.2	2.0	2.0	2.3	—	—	—
One	150	2.5	2.4	2.5	—	—	—	—	—
One	205	3.0	—	—	—	—	—	—	—
Two	10	0.1	*	*	*	*	*	*	0.1
Two	25	0.2	0.1	*	*	*	*	0.1	0.2
Two	50	0.4	0.2	0.1	*	*	0.1	0.2	*
Two	100	0.7	0.5	0.3	0.7	1.0	—	—	—
Two	150	1.0	0.7	1.2	—	—	—	—	—
Two	205	1.5	—	—	—	—	—	—	—

Amprobe[®] Transducer Specifications

AC Current Models

Note: If specifications of a particular transducer do not satisfy your particular application, contact factory for possible modifications. For applications involving 400Hz, contact factory.

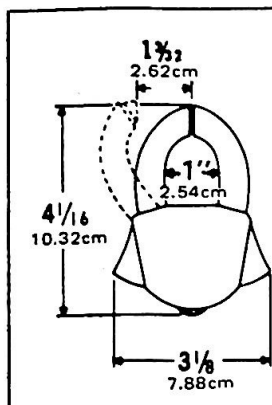


Figure 1

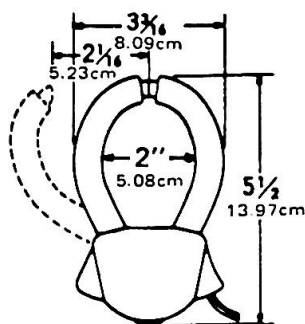


Figure 2

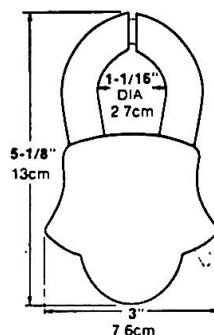


Figure 3

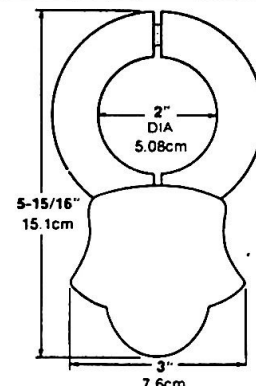


Figure 4

MODELS A-60FL, A-70FL, A663-1, A664-1 ††

	A-60FL (Fig. 1 and 13)	A-70FL (Fig. 2 and 13)	A663-1+ (Fig. 3)	A664-1+ (Fig. 4)
Current Ranges	0-5/25/100/250 Amps AC	0-10/50/250/500 Amps AC	0-300 Amps AC	0-1000 Amps AC
Secondary Output	0-260μ ADC into 400Ω load on each range*	0-260μ ADC into 400Ω load on each range*	1mV AC output per 1 Ampere AC input**	1mV AC output per 1 Ampere AC input**
Accuracy based on Sinusoidal Waveform	±2% F.S.	±2% F.S.	±1% of reading	±1% of reading
Frequency	60 Hz (for 50 Hz A-65FL)	60 Hz (for 50 Hz A-75FL)	50 - 400 Hz	50 - 400 Hz
Breakdown Voltage (between secondary & core)	1500 Volts RMS	1500 Volts RMS	1500 Volts RMS	1500 Volts RMS
Case Voltage Breakdown (between laminations and case)	3000 Volts RMS (4200 Peak)	3000 Volts RMS (4200 Peak)	3000 Volts RMS (4200 Peak)	3000 Volts RMS (4200 Peak)

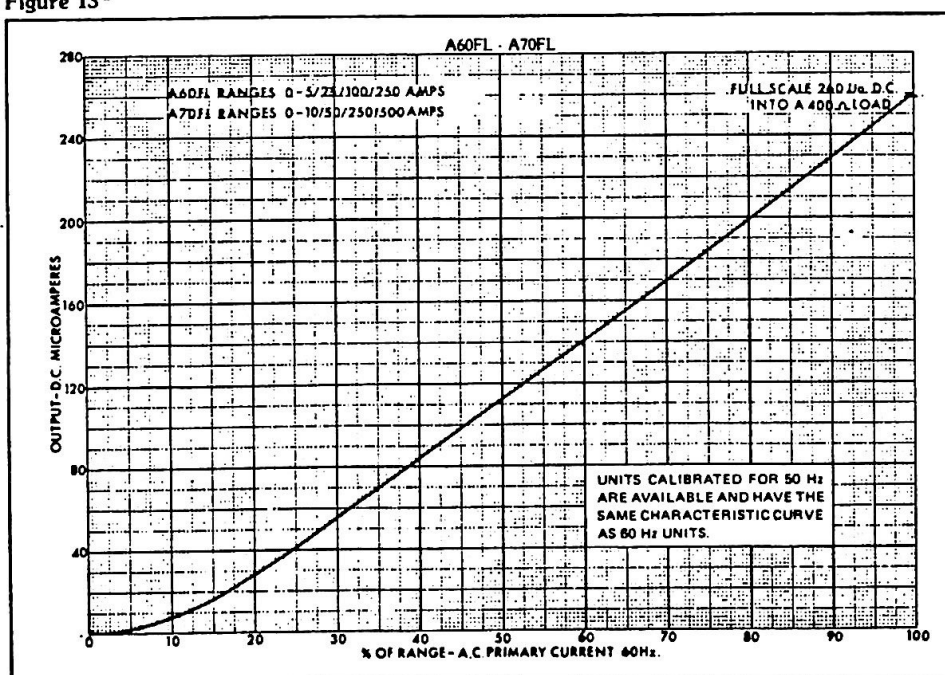
†† Other variations including 50Hz and 400Hz available with and without line cord and or circuit board.

Note: A-60FL A-70FL have rotary range switches in base of transducer body. All of above units supplied with line cord 4 1/2 to 5 feet long.

*See curves Fig. 13**Minimum input impedance measuring instrument 10,000 ohms.

+ These models do not have output curves shown, since their output is very linear.

Figure 13*



APPENDIX E-1

ADC-1 TECHNICAL SPECIFICATIONS

- * 16 Analog to Digital Inputs
 - * Integrating dual slope A/D with conversion at 7 or 14 Hz.
(30 Hz. with clock speed jumper and two capacitor changes)
 - * 12 bit (plus sign) A/D provides resolution of one part in 8192.
(0.1 mV sensitivity).
 - * Input Range - (modifiable from +0.4 volts to +2.0 Volts)
 - * Maximum Input Voltage +5 volts.
 - * Common Mode Rejection Ratio 50uV/V.
 - * Optional instrumentation amplifier allows microvolt sensitivity for one bank of 8 inputs.
- * 4 Digital Inputs
 - * 2 modes - toggle or latched
 - * maximum sample rate @ 9600 baud is 400 Hz.
 - * functions for both normally open and normally closed circuits.
- * 6 Direct Wire Controlled Outputs
 - * Latched TTL drivers with maximum current of 50 mA.
 - * Maximum voltage - 5 volts.
- * AC Line Carrier Remote Device Control
 - * sends signals over normal household/office AC wiring to control 32 independent devices using BSR, Leviton, Sears or Radio Shack remote control modules.
 - * 8 House codes are switch selectable to reach 256 possible units.
 - * Signal frequency 120kHz, 1msec pulse duration, 8 volt amplitude.
 - * Code transmission time - 0.4 sec.
- * RS-232C or serial interface connection to most computers or modems.
 - * 6 switch selectable baud rates - 300,600,1200,2400,4800,9600.
 - * Uses three wires from DB-25 for easy connection to your system.
 - * Direct TTL serial connection for Atari, Commodore VIC 20/64.
 - * Requires no parity - uses full 8 bits for data.
 - * Stop Bits - 2; will function with one.
 - * Bit mapped communications; byte returned for each byte received.
 - * Select/disable line for multiple unit operation.
- * Power Requirements - 20mA @ 5 v. or 35mA @ 12 v. (selectable on DB-25).
 - * Power can generally be supplied by host computer/modem.
 - * CMOS technology used throughout for low power consumption
 - * No heat generation.
- * Operating Temperature - 0°C(32°F) to 50°C(122°F).
- * Dimensions - ADC-1 desktop unit - 8x7x2 inches, aluminum enclosure.
 - BSR AC signal transmitter - 3.5x2.5x1.5 inches
- * Weight - 2 lbs.

APPENDIX E-2

PIN ASSIGNMENTS FOR THE ADC-1 DB-25 CONNECTOR

<u>PIN</u>	<u>SIGNAL</u>	<u>IN/OUT</u>
2	Transmitted data	RS232-out
3	Received data	RS232-in
7	Signal Ground	
8	Board select	Deselect by grounding
9	Power Input	+6V to +24V
11	BSR Transmitter	
12	BSR Transmitter	
13	BSR Transmitter	
14	Controlled Outputs Power Input	Requires +5 v. shipped with 600 ohm to +5 V (pin 18).
18	Power Input	+5V or current limited +12V
21	Received Data	TTL-in
22	Transmitted Data	TTL-out

Note: Either pin 9 or pin 18 may be used to input power to the ADC-1.
Connection to both is not required.

Power input

Radio Shack AC adapter

9 VDC at 300 ma

pin 9 (+6v to +24v)

pin 7 (ground)

APPENDIX E-3

PARTS LIST

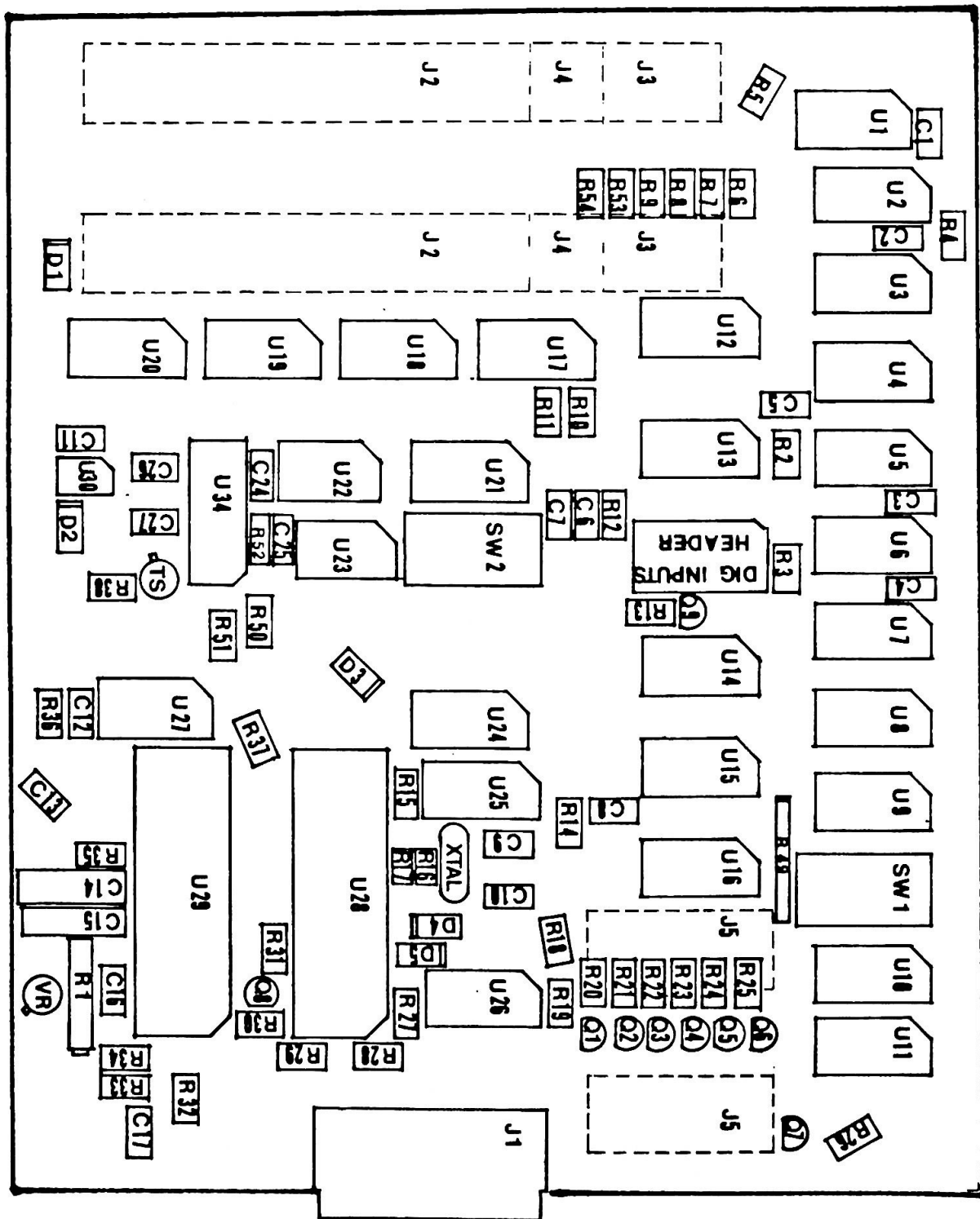
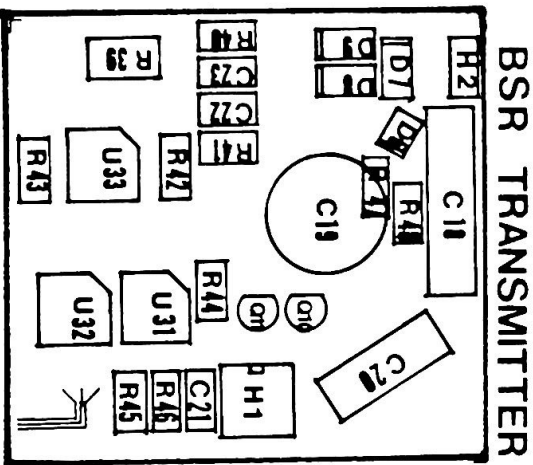
COMPONENT CODE	FUNCTION	DESCRIPTION
U1,U3,U12,U13,U22	Digital Inputs, BSR	4013
U2,U6,U21	A/D Delay, BSR timing	4528
U4	BSR misc	4011
U5,U9,U25	Clock, BSR Control	4069
U7,U24	Baud Rate, BSR status	4020
U8,U10,U117	BSR Shift registers	4014
U14	Data MUX for Dig In	4502
U15,U16	Output control	74C174
U17,U18,U19,U20	Analog Multiplexors	4051
U23	Instruction decode	4556
U26	RS232 Signal	LM324
U27	Analog multiplexer	4052
U28	UART	IM6402IPL
U29	12 BIT A/D Converter	ICL7109CPL
U30	-5 Voltage Regulator	ICL7660
U31,U32	6 pin optoisolator	4N33
U33	BSR AC Mod timing	ICL7555IPL
U34	Instrumentation amp	ICL7605CJN
C1,C2	ceramic	.01uf-25v
C4,C5	ceramic disk	.1 uf-100v
C3	radial	.22uf-12v
C6,C23	ceramic disk	.01uf-50V
C7,C8	ceramic monolithic	1.0uf-10v
C9,C10	ceramic	39 pf-50v
C11,C13,C16,C17	tantalum electrolytic	10 uf-10V
C12	ceramic	.001uf-1KV
C14	A/D Integrator-polypropylene	.33uf-100V
C15	A/D Integrator-polypropylene	.15uf-100V
C18	non-polar	2.2uf-160v
C19	electrolytic	1000uf-25v
C20	non-polar	.22uf-160V
C21	ceramic disk	.0016uf-1000
C22	ceramic disk	220pf-500v
C24,C25,C26,C27	high gain amp; non-polar	1 uf-10V
C28-C35	underchip filter	.03uf-12v
C36	A/D underchip filter	.03uf-12v
R1	A/D adjust - linear 50ppm/C	2K TRIM
R2,R3,R39,R50	BSR- linear 50ppm/C	50K TRIM
R4,R5,R10,R12,R28,R31,R32,R41,R47		10K -1/4W
R6,R7,R8,R9,R11,R14 R15,R27,R30,R37		24K -1/4W
R13		470 -1/4W
R16		1meg -1/4w
R17,R20-R25		300 -1/4W
R18,R19		51K -1/4W
R26,R34,R42		5.6K -1/4W

REMOTE MEASUREMENTS SYSTEMS, INC. - PARTS LIST - ADC-1

COMPONENT CODE	FUNCTION	DESCRIPTION
R33	<=150ppm/c temp sens	6.81K -1/4W
R36		100K -1/4W
R38, TEMP	1% metal film	1K -1/4W
R40,R35		20K -1/4W
R43		2.2k -1/4W
R44		20 -1/4W
R45		1K -1/4W
R46		15K -1/4W
R29		200 -1/4W
R48		350K -1/4W
R49	7 Resister pack-pullups	10 K pack
R51	1% metal film	50K -1/4W
R52	gain determine inst amp	950K -1/4W
D1	Main Power 5.6v Zener 1/2W	1N5232BTR
D2,D3,D4,D5,D6	RS232, -5v, UART diodes	1N4148
D7,D8	BSR 1 amp	1N4007
D9	BSR 18v zener 1W	1N4746A
H1	BSR transformer	KC1025
H2	Filter coil - BSR	200uH
Q1-Q7,Q11,Q12	Output drivers + BSR	PN2222
Q8,Q10	Status gate, reset	PN2907
Q9	A/D voltage Reference 50ppm/C	AD589KH
S1,S2	House Select & Baud Rate	8 pos dipsw
S3	Digital inputs header	16 pin 2x8
TS1	Onboard temperature sensor	AD590JH
X1	Crystal	2.4576 MHz

APPENDIX E-4 COMPONENT LAYOUT DIAGRAM

ADC-1



APPENDIX F

REFERENCES

- Ciarcia, Steve, 1981. "Ciarcia's Circuit Cellar," BYTE/McGraw-Hill, Peterborough, N.H.
- Cooper, G.R. and C.D. McGillem, 1967. "Methods of Signal and System Analysis", Holt, Reinhart and Winston; New York.
- Finkel, A., N. Harris, P. Higginbottom, and M. Tomczyk, 1982, "VIC 20 Programmer's Reference Guide", Commodore Business Machines and Howard W. Sams, Indianapolis, Indiana, 289 pg.
- Fritschen, Leo J., and Gay, Lloyd W. 1979. "Environmental Instrumentation," Springer-Verlag, New York.
- Johnson, Roger, 1983. "Designing with Optical Shaft Encoders", Robotics Age, May/June, pg 27-34.
- Marston, R.M., 1977. "110 Electronic Alarm Projects," Hayden Book Company, Rochelle Park, New Jersey.
- National Semiconductor Corporation, 1977. "Pressure Transducer Handbook," Santa Clara, CA.
- Sargent, Murray and Shoemaker, Richard L. 1981. "Interfacing Microcomputers to the Real World," University of Arizona, Addison Wesley Publishing Company, Reading, Massachusetts.
- Titus, Jonathan A. and Titus, Christopher A. and Larsen, David-G., 1980. "Book 2, TRS -80 Interfacing," Howard W. Sams & Co., Inc., Indianapolis, Indiana.

1 REM Program to read ADC-1 temperature - for Radio Shack Model 100

```
5   OPEN "COM:88N2D" FOR INPUT AS 1   ;set baud rate,bits/byte
10  PN=192                             ;serial port address
20  CLS                                 ;clear screen
50  CN=24                               ;set to read onboard temp
60  GOSUB 5000                         ;goto channel read subroutine
                                       ;return will have millivolts*10
70  DK= Z/10                           ;calc temp in Kelvin = millivolts
80  DF=(9/5*(DK-273))+32               ;convert to degrees Fairenheit
90  PRINT @120,"TEMP= ";DF;           ;print temp 4 lines down
100 GOTO 50                            ;do it again

5000 OUT PN,CN                         ;output to port A/D start
5010 GOSUB 5200                        ;wait for response
5020 OUT PN,161                        ;get high byte
5030 GOSUB 5200                        ;wait for response
5040 HB=CH                             ;store high byte
5050 IF (HB AND 128)<>0 THEN 5020      ;check for A/D not done
5060 OUT PN,145                        ;if done, get low byte
5070 GOSUB 5200                        ;wait for response
5080 LB=CH                             ;store low byte
5085 HM=HB AND 15                      ;retain 4 bits of high byte
5090 Z=LB+256*HM                       ;combine high and low bytes
5100 IF (HB AND 16)=0 THEN Z=-Z        ;fix sign (polarity)
5110 RETURN                            ;done

5200 FOR DE=1 TO 25: NEXT              ;delay to wait for response.
                                       ;delay time should be optimized
                                       ;for baud rate. Set to 1 for
                                       ;baud rates >= 4800.
5210 CH=INP(PN)                        ;get ADC-1 response
5220 RETURN                            ;done
```



```

1  REM Program to monitor the digital inputs on Remote Measurement
2  REM Systems' ADC-1 using a Radio Shack Model 100.
3  REM Keith Ronnholm, 2/2/84, 206-525-3369.

5  OPEN "COM:88N2D" FOR INPUT AS 1      ;set baud rate,bits/byte
10  PN=192                               ;serial port address
20  CLS                                  ;clear screen
25  CT=0                                 ;initialize counter
30  GOSUB 6000                           ;get status of digital inputs
40  PRINT @0,"Digital input number:"    ;print heading
52  FOR I=1 TO 4                         ;loop for each input
53  DT=2^(I-1)                           ;select bit to check
54  CP=80+(I-1)*10                       ;set horizontal display position
55  PRINT @CP,"#":PRINT @(CP+1),I;       ;print input number
56  IF DT AND D THEN PRINT @(CP+40),"ON"; ELSE PRINT @(CP+40),"OFF";
59  NEXT                                ;print ON or OFF for each input
60  CT=CT+1                              ;increment counter
65  PRINT @311,CT;                       ;print number of times all
100 GOTO 30                              ;four inputs have been checked

5200 FOR DE=1 TO 25: NEXT                ;delay to wait for response.
                                           ;delay time should be optimized
                                           ;for baud rate. Set to 1 for
                                           ;baud rates >= 4800.
5210 CH=INP(PN)                          ;get ADC-1 response
5220 RETURN                              ;done

6000 OUT PN,129 : GOSUB 5200              ;read inputs without resetting
6010 D=CH AND 15                         ;mask for digital input bits
6020 OUT PN,128 : GOSUB 5200             ;reset latched inputs
6030 RETURN                             ;return to calling program

```



```

1  REM Program to change the controlled outputs on Remote Measurement
2  REM Systems' ADC-1 using a Radio Shack Model 100.
3  REM Keith Ronnholm, 2/2/84, 206-525-3369.

4  DIM DS(6)                                ;output status array
5  OPEN "COM:88N2D" FOR INPUT AS 1          ;set baud rate,bits/byte
10  PN=192                                  ;serial port address
20  CLS                                     ;clear screen
30  GOSUB 8000                             ;turn all ouputs off (DS=0 at start)
52  FOR I=1 TO 6                           ;loop to print each output state
54  CP=40+(I-1)*7                          ;set horizontal display position
55  PRINT @CP,"#":PRINT @(CP+1),I;         ;print output number
56  IF DS(I)=1 THEN PRINT @(CP+40)," ON"; ELSE PRINT @(CP+40),"OFF";
59  NEXT                                    ;print ON or OFF for each input
60  PRINT @160," "                          ;set position for Inputs
70  INPUT "Enter controlled output number:";J
80  INPUT "Enter 1 for ON or 0 for OFF";DS(J)
85  IF DS(J)=1 OR DS(J)=0 THEN GOTO 100    ;check for illegal input
87  PRINT "ERROR ":BEEP:GOTO 80            ;tell user if illegal
100 GOTO 20                                ;send new output state, ask for
                                           ;next change

5200 FOR DE=1 TO 25: NEXT                 ;delay to wait for response.
                                           ;delay time should be optimized
                                           ;for baud rate. Set to 1 for
                                           ;baud rates >= 4800.

5210 CH=INP(PN)                           ;get ADC-1 response
5220 RETURN                                ;done

8000 DT=DS(1)+2*DS(2)+4*DS(3)+8*DS(4)+16*DS(5)+32*DS(6)+64
                                           ;place states into proper bits
8010 OUT PN,DT : GOSUB 5200               ;send command byte
8020 RETURN                                ;return to calling program

```



```

1  REM Program to control BSR remote control modules via
2  REM Remote Measurement Systems' ADC-1 and a Radio Shack Model 100.
3  REM by Keith Ronnholm, 2/2/84

4  DIM DV(16)                                ;BSR device code array
5  OPEN "COM:88N2D" FOR OUTPUT AS 1          ;set baud rate for port
10  PN=192                                    ;set I/O port
50  FOR K=1 TO 16                            ;set first house BSR codes
60  READ DV(K)                              ;from DATA @ line 7900
70  NEXT K
75  CLS                                      ;clear screen
80  INPUT "WHAT BSR MODULE NUMBER SHOULD I CONTROL? ";BM
90  INPUT "SHOULD I TURN IT ON (1), OFF(2), OR MAKE IT COUNT TO 5 (3)";AC
100 UC=DV(BM)                               ;select device code
110 GOSUB 7000                              ;send address code to AC line
120 IF AC=1 THEN UC=197 :GOSUB 7000         ;if 1 then send an ON
130 IF AC=2 THEN UC=199 :GOSUB 7000         ;if 2 then send an OFF
140 IF AC=3 THEN GOSUB 180                  ;if 3 then make lights flash
150 GOTO 75                                ;go ask for next action

180 UC=199                                  ;begin count to five loop. make
                                           ;sure light is off before counting
190 GOSUB 7000                              ;send OFF to unit
200 FOR J= 1 TO 5                          ;count loop
210 FOR I=1 TO J                            ;loop to send flashes
220 UC=197                                  ;turn unit ON
230 GOSUB 7000                              ;send ON command
240 UC=199                                  ;turn unit OFF
250 GOSUB 7000                              ;send and wait
260 NEXT I                                  ;end flash loop
270 FOR K=1 TO 200:NEXT                    ;wait before next # of flashes
280 NEXT J                                  ;end count loop
290 RETURN

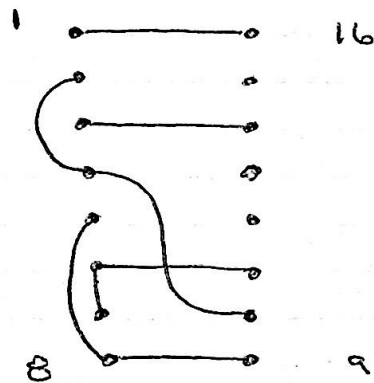
5200 FOR DE=1 TO 25:NEXT                  ;delay to wait for response byte
                                           ;delay should be optimized for
                                           ;baud rate being used.
5210 CH=INP(PN)                            ;get ADC-1 response
5220 RETURN                                ;return to calling prog

7000 OUT PN,UC                             ;start ADC-1 transmitting code
7010 GOSUB 5200                            ;wait for ADC-1 response
7020 OUT PN,129                            ;get BSR busy/done status
7030 GOSUB 5200                            ;wait for answer
7040 IF (CH AND 16)<>0 THEN 7020            ;if still busy, ask again
7050 RETURN                                ;done, ret main program

7900 DATA 204,220,196,212,194,210,202,218,206,222,198,214,192,208,200,216
                                           ;module addresses for BSR units

```


Original header



Channel #1

data - + 5 volts

(4 - 10)

latched

reset #1 - reset

(6 - 11)

manual reset

Channel #2

data - output (for #2)

(3 - 14)

toggle

reset - ground

(8 - 9)

(auto-reset)

Channel #3

data - +5 volts

(2 - 10)

latched

reset #3 - reset

(7 - 11)

manual reset

Channel #4

data - output

(1 - 16)

toggle

reset - ground

(5 - 9)

(auto-reset)

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

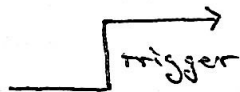
100

Original configuration

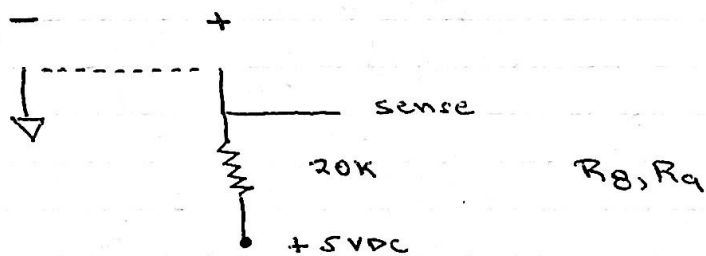
Digital inputs

switch	NC/NO	L/T
1	NC	L
2	NC	T
3	NO	L
4	NO	T

All inputs trigger on rising edge



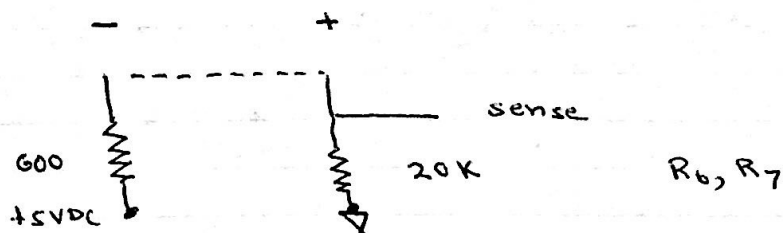
normally closed 1 + 2



closed → open



normally open 3 + 4



open → closed



